INTERNET-DRAFT
<draft-yu-imap-client-id-03.txt>
Intended Status: Standards Track
Expires May 19, 2020

Y. Deion
LinuxMagic

November 19, 2019

IMAP Service Extension for Client Identity
<draft-yu-imap-client-id-03.txt>

Abstract

This document defines an Internet Message Access Protocol (IMAP) service extension called "CLIENTID" which provides a method for clients to indicate an identity to the server.

This identity is an additional token that may be used for security and/or informational purposes, and with it a server may optionally apply heuristics using this token.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The [IMAP] protocol and its extensions describe methods whereby an client may provide identity and/or authentication information to an IMAP server. However, these existing methods are subject to limitations and none offer a way to identify the IMAP client with absolute confidence. This document defines an IMAP service extension to provide an additional identity token which can represent the IMAP client with a higher degree of certainty when accessing the IMAP server.

Typically IMAP clients enter the authenticated state by using either the AUTHENTICATE or LOGIN command. IMAP servers are often subject to malicious clients attempting to use authorization credentials and/or identities not intended for their use (e.g. stolen credentials or brute force attacks). When such an attack is attempted, the IMAP server may be unable to identify the impersonation and restrict such an unintended use by someone other than the authorized user or said credentials. While there are ways to identify the source of the IMAP client such as its IP address, it would be useful if there was an additional way to uniquely identify the client in a method solely available across an encrypted channel.

Using the CLIENTID extension, an IMAP client can provide an additional identity token to the server called its "client identity".

The client identity can provide unique characteristics about the client accessing the IMAP service and may be combined with existing identification mechanisms in order to identify the client. An IMAP server may then apply additional security policies using this identity such as restricting use of the service to clients presenting recognized client identities or only allowing use of authorized identities that match previously established client identities.

The CLIENTID extension is present in any IMAP implementation that returns "CLIENTID" as one of the supported capabilities to the CAPABILITY command.

2. Conventions Used in This Document

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 [KEYWORDS].

Formal syntax is specified using [ABNF].

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server.

"Connection" refers to the entire sequence of client/server interaction from the initial establishment of the network connection until its termination.

3. CLIENTID

3.1. CLIENTID Command

```
Arguments:  client identity type
            client identity token

Responses:  no specific responses for this command

Result:     OK - clientid completed, client identity stored
            BAD - command unknown or arguments invalid
```

Note that a valid CLIENTID command will never return the NO result because heuristics MUST NOT be applied to the CLIENTID arguments at this stage. Instead the client identity information SHOULD be stored and passed along to any and all [SASL] authentication mechanisms.

3.2. CLIENTID Arguments

The CLIENTID command takes the following two arguments:

1. client identity type: A string identifying the identity type the client is providing. It MUST be between 1 and 16 characters and comprised of only alphanumeric and dash characters.

2. client identity token: A string identifying the client. It MUST be between 1 and 128 printable characters.

The IMAP server MUST reject any CLIENTID command with badly formatted arguments. The IMAP server MUST accept the arguments from a valid CLIENTID command and SHOULD store it at the minimum for the remaining duration of the IMAP connection.

3.3. Advertising the CLIENTID capability

The CLIENTID capability is used to tell the IMAP client that the IMAP server supports the CLIENTID extension. However, certain conditions MUST be met before the IMAP server advertises the CLIENTID capability.

1. The IMAP server and IMAP client MUST negotiate encryption via STARTTLS/SSL or some other secure mechanism.

2. The IMAP server MUST be in the non-authenticated state.

3. The IMAP server MUST have the CLIENTID extension support enabled.

While all the conditions are met, the IMAP server MUST advertise the CLIENTID capability in all proceeding CAPABILITY commands.

3.4. Restrictions on the CLIENTID command

Under certain circumstances, the use of the CLIENTID command will be restricted:

1. Before the CLIENTID capability has been advertised, the IMAP server MUST reject any issued CLIENTID commmand and the IMAP client MUST NOT issue the CLIENTID command.

2. Outside of the non-authenticated state, the IMAP server MUST reject any CLIENTID command issued by the IMAP client and the IMAP client MUST NOT issue the CLIENTID command.

3. Once a valid CLIENTID command has been issued, the IMAP server MUST reject any further CLIENTID command issued by the IMAP client and the IMAP client MUST NOT issue any subsequent CLIENTID commands.

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form notation as specified in [ABNF]. [IMAP] defines the non-terminals "capability" and "command-nonauth".

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
capability      =/ "CLIENTID"

command-nonauth =/ client-id
```

```
   client-id       = "CLIENTID" SP client-id-type SP client-id-token

   client-id-type  = 1*16 ALPHA / DIGIT / "-"
                   ;; alphanumeric with dash character

   client-id-token = 1*128 VCHAR
                   ;; any printable US-ASCII character
```

5. Discussion

5.1. Applying heuristics to CLIENTID

This section discusses the possible heuristics that can be applied to the information that is presented via the CLIENTID command. This information includes whether a valid CLIENTID command was issued, the client identity type and the client identity token.

1. The IMAP server MAY choose to require that a successful CLIENTID command be issued or that a particular client identity type be presented before processing or accepting an authentication request.

2. The IMAP server MAY reject any authentication request not preceded with a client identity type that matches ACL’s or rules as defined in the IMAP server.

3. An IMAP server MAY reject any authentication request preceded by a CLIENTID command that contains a client identity type or client identity token that the server chooses not to accept for any reason such as by policy.

4. An IMAP server MAY reject any authentication request preceded by a CLIENTID command that contains a client identity type or client identity token that the server has chosen to disable or revoke use of either temporarily or permanently.

The IMAP server SHOULD only ever reject an IMAP client based on CLIENTID information during or after the authentication process/handler. In the interest of limiting the amount of information being revealed, the rejection message SHOULD be as generic as possible and SHOULD NOT reveal any information on the heuristics.

Even if the client identity type and/or client identity token are not recognized, supported or permitted by the server and/or the owner of the authentication credentials, the presented information may still be useful for analysis.

5.2. Utility of CLIENTID

Regardless of how frowned upon, users commonly reuse authorization information (like the username and password pair) across multiple services. When one service is compromised, malicious actors can also gain access to other services where the user also used the same credentials. Based on this representative problem alone, the utility of CLIENTID as an additional layer of determining the rights to present such authorization information becomes quickly apparent.

The utility of CLIENTID may be seen by considering the following:

1. An IMAP server could recognize a device not historically known to have presented the authentication credentials before.

2. An IMAP server could restrict authentication from actors not presenting a valid CLIENTID, or an account holder that the IMAP server provides service for could restrict authentication to only those devices that present valid CLIENTID.

3. An IMAP server could restrict authentication to only devices which present a CLIENTID containing a client type identifier which the account holder or operator of the server deems to be permitted. (Eg. Only allow vendor A’s devices)

4. An IMAP server could alert an account holder that an attempt to present their authorization credentials came from an unknown, unrecgonized, or different device.

However, this extends beyond just the restriction of authentication. While it might be argued that this can be served as a special form of SASL, by implementing this in the IMAP service itself, the IMAP service can choose before allowing a connection to be passed to a SASL implementation, allowing it to perform other heuristics, such as brute force attacks, more effeciently.

Recent evolution of the internet as a whole, has brought about large scale data breaches, compromised botnets comprising of millions of nodes, and transitions to Carrier Grade NAT, and the flourishing of IoT devices, means traditional methods of protecting against brute force attacks have become much more difficult. Traditional methods such as rate limiting and/or blocking access by IP are no longer viable without introducing collateral effects, such as either blocking legitimate users, or creating the conditions that allow DOS (Denial of Service) to legitimate users.

Historically, IMAP and other services used what is technically a two factor, the email/user and password, albeit not effective in that the email is a KNOWN value. And with the propensity for users to use a simple password, and the hundreds of millions of email addresses exposed in data breaches, or available by other means the ability to brute force is quite simple. While of course it is recommended that users use longer and more secure passwords, this is not the de facto situation, and the threats when credentials get compromised are significant. And with ’botnet’ operators able to engage millions of IoT in a distributed brute force, the status quo is dangerous. Adding another non-public factor to be used as part of access control adds a strength against brute force by many factors. This accomplishes that in a backwards compatible fashion, encouraging adoption.

But for the IMAP server, it also offers additional abilities.

Historically, under brute force attacks, rate limiting or blocking by IP Address was possible with little damage. But with the proliferaction of IoT devices, smart phones, and the run-out of IP space, we have conditions where thousands of devices could be behind an IP Address, or IP(s) that are dynamic with devices changing IP(s) in minutes, blocking or rate limiting an IP bears risks of blocking legitimate users. By implementing a level of uniqueness to a connecting device, introduces the ability to restrict or block a subset from connecting to the service for either brute force or dictionary attacks, while still allowing other devices to continue to be able to present authentication successfully.

While 'forgery' and/or the use of random client identifier is possible, such behavior is also more readily detectable when a device identifier is presented.

1. The IMAP server, when faced with hundreds of devices behind the same IP address, during an attack can restrict authentication attempts to only connections presenting a valid client identifier token.

2. The IMAP server, during an attack, can restrict authentication to only historically known devices.

3. The IMAP server can differentiate between many different devices behind the same IP, and apply maximum connections per device, rather than maximum connections per IP.

4. While a person may present authentication credentials from many different geographical locations, eg, home, office, and travel, a single device will not in general be able to be in two geographical locations at the same time. The IMAP server will have new information to apply to threat detection heuristics, ie to treat the use of the same client indentifier token from two locations, as a possible brute force or forgery situation.

5.3. Use Cases of CLIENTID

With CLIENTID the IMAP server has additional information it may use in its interactions with the client. It may:

1. Restrict use of an authorization tokens to a set of client identity token identities, thereby offering an added level of security. For example the use of authorization credentials may only be accompanied by a specified set of CLIENTID tokens and/or types for a specific account holder, or set of account holders

2. Identify that the same CLIENTID token is used to access multiple authorized identities, and restrict access to the IMAP service. For example a malicious client that has attempted to gain access using multiple authorization tokens may be identified through its unusual behavior.

3. Retain knowledge of CLIENTID tokens previously presented with

specific authorization credentials, and if the token has not been previously seen, restrict access to the IMAP service.

4. Require that the IMAP client present a token such as a license key established outside of the IMAP session in order to make use of any authorized identity.

5. Apply different security policies to clients that provide a CLIENTID token versus those which do not. For example, provide clients providing such an identity with additional trust.

6. Ability to rate limit or block based on the presented client-identifier-token, when multiple devices use a shared IP address, without affecting other devices.

7. Ability to detect distributed and localized dictionary attacks and brute force attacks.

8. Use the client-indentifier-token as a third factor to be passed to authentication methods. [SASL]

5.4. Other IMAP Client Identifiers

The [IMAP] protocol and its extensions describe methods whereby an

IMAP client may provide identity information to an IMAP server. Some of these identitiers are listed for contrast:

1. The client connection provides a source IP address associated with the IMAP session. This may be accompanied by a PTR record and/or GeoIP information.

2. The AUTHENTICATE and LOGIN command allows the client to present a user and/or password/authentication mechanism for an IMAP session.

5.5. Future Considerations

In the future there may be a demand for being able to provide multiple CLIENTID commands with different client identity types. For instance, it may be desirable for a device to indentify itself, both with a hardware device identifier, and a software identifier. We believe this to be out of scope, and can be accomodated with a special client-identifier-token which encapsulates both.

6. Client Identity Types

This document does not specify any CLIENTID identity type that MUST be supported. The client identity type is meant to be defined by the client implementation that is designed to access the IMAP server and protocol. For instance, many IMAP client software implementations already create a distinct UUID for each account. Some commercial email clients have a license key. Some physical devices that need to of client identity type that conforms to the definition, it is interact with IMAP might have a unique hardware ID or MAC Address. While there is no pre-defined list of client identity type defined by this RFC, and all IMAP servers should be prepared to accept any form suggested that IMAP client developers carefully consider the name of the client identity type. For example, rather that using a client identity type of UUID, consider the advantages of making it more distinct, eg "<product_short_code>UUID". This way the IMAP server can better record histories, eg the difference between say a Thunderbird generated unique id, and a Mutt generated unique id.

Some examples of identity type might be UUID, LICENSE, DEVICE_ID, MAC and/or COOKIE. It is expected that the most common types might be related to distinct UUID, LICENSEKEY, or HARDWAREID.

An IMAP server SHOULD NOT reject an unidentified CLIENTID type, except for specific policy use cases.

It is envisioned that in the future it will be useful to propose a set of standardized client-indentity-type to help with validation, or to allow the IMAP server to apply ACL rules on expected types, this would be an extension to this RFC.

1. UUID

   UUID is a common practice to represent either a individual user, hardware device or software installation associated with a specific individual. The support of UUID enables existing UUID implementations to be used to semi-uniquely identify a device associated with an individual. A definition of the format should be considered. Otherwise non-standard UUID might be a separate type specific to the software implementation, for instance TBIRD-UUID.

2. LICENSE

   An IMAP client may find it useful to identify the license key of software it is using. Such licenses are typically crafted such that they are unique and useful to identify a software installation. This is more normally suited for a software designed for a single-user. While LICENSE could be standard type again, it might more more helpful to specify a vendor specific type such as BBLICENSEKEY.

3. DEVICE_ID

   Many hardware devices are designed to be used by a single individual and already have an associated hardware device id. While a standard type might be defined, it also might be more helpful to use a vendor specific type, such as ATOM-DEVICEID.

4. MAC

   The MAC address traditionally was used as a worldwide identifier both of the unique device, as well as it’s vendor and product category, however this is not always the case anymore, in the case of it’s usage in ’virtual’ devices. But for many hardware devices which are required to access a defined IMAP resource, the MAC address may still be a simple unique identifier. MAC should NOT be used, unless this is a MAC address that can be associated to a vendor using standard MAC registration information as defined or set by the IEEE Standards Association and is meant to represent a unique device.

5. COOKIE

While not guranteed to be consistent many web applications are designed to access IMAP directly and may need to have a semi-unique identifier available as part of the web based transaction. It is assumed that COOKIE encompasses the group of web based tokens known to persist from session to session. A specific web based application can provide sufficient information in the actual client-identifier-token to differentiate between applications and or websites, and are convenient as they can be related to very specific domains, and are universally available to web application designers.

As a reminder, an IMAP server SHOULD NOT retain and/or store the CLIENTID information WITH authentication credentials or authentication systems directly, but the IMAP service MAY associate the CLIENTID with a specific account holder, eg to create a history file of known CLIENTID tokens associated or permitted to access or present authentication credentials for that account holder.

This document recommends that an IMAP server handle any given client identity type from a CLIENTID command in one or more of the following manners.

1. Handled but treat as not presented (ignored, no persistance)
2. Store in IMAP session but treat as not presented (debugging)
3. Store in the IMAP session, so it is available to System log
4. Store in the IMAP session, so it is available to User log
5. Use for authentication
6. Use for alert when authentication fails
7. Use for alert when authentication succeeds
8. Unused

7. Examples

7.1. UUID as Client Identity

```
C: [connection established over a plaintext connection]
C: a001 CAPABILITY
S: * CAPABILITY IMAP4rev1 STARTTLS AUTH=GSSAPI LOGINDISABLED
S: a001 OK CAPABILITY completed
C: a002 STARTTLS
S: a002 OK STARTLS completed
<TLS negotiation, further commands are under [TLS] layer>
C: a003 CAPABILITY
```

```
   S: * CAPABILITY IMAP4rev1 AUTH=GSSAPI AUTH=PLAIN CLIENTID
   S: a003 OK CAPABILITY completed
   C: a004 CLIENTID UUID 23bf83be-aad7-46aa-9e0f-39191ccf402f
   S: a004 OK CLIENTID completed
   C: a005 LOGIN joe password
   S: a005 OK LOGIN completed
```

7.2. Malformed CLIENTID Command

```
   C: [connection established over a plaintext connection]
   C: a001 CAPABILITY
   S: * CAPABILITY IMAP4rev1 STARTTLS AUTH=GSSAPI LOGINDISABLED
   S: a001 OK CAPABILITY completed
   C: a002 STARTTLS
   S: a002 OK STARTLS completed
   <TLS negotiation, further commands are under [TLS] layer>
   C: a003 CAPABILITY
   S: * CAPABILITY IMAP4rev1 AUTH=GSSAPI AUTH=PLAIN CLIENTID
   S: a003 OK CAPABILITY completed
   C: a004 CLIENTID UUID
   S: a004 BAD Error in IMAP command received by server
```

The IMAP server rejects the CLIENTID command as it is not well formed due to there being only a single parameter provided.

7.3. Client Identity without TLS/SSL Session

```
   C: [connection established over a plaintext connection]
   C: a001 CAPABILITY
   S: * CAPABILITY IMAP4rev1 STARTTLS AUTH=GSSAPI LOGINDISABLED
   S: a001 OK CAPABILITY completed
   C: a002 CLIENTID UUID 23bf83be-aad7-46aa-9e0f-39191ccf402f
   S: a002 BAD Unknown IMAP command received by server
```

The IMAP server rejects use of the CLIENTID command as the CLIENTID capability had not been advertised because no encryption was negotiated between the IMAP server and IMAP client.

7.4. Client Identity Leading to Rejection

```
   C: [connection established over a plaintext connection]
   C: a001 CAPABILITY
   S: * CAPABILITY IMAP4rev1 STARTTLS AUTH=GSSAPI LOGINDISABLED
   S: a001 OK CAPABILITY completed
   C: a002 STARTTLS
   S: a002 OK STARTLS completed
   <TLS negotiation, further commands are under [TLS] layer>
   C: a003 CAPABILITY
   S: * CAPABILITY IMAP4rev1 AUTH=GSSAPI AUTH=PLAIN CLIENTID
   S: a003 OK CAPABILITY completed
   C: a004 CLIENTID UUID 23bf83be-aad7-46aa-9e0f-39191ccf402f
   S: a004 OK CLIENTID completed
   C: a005 LOGIN joe password
   S: a005 BAD Failed to authenticate
```

The IMAP server rejects use of the system during the LOGIN command after deciding that the provided client identity does not establish sufficient privileges. Note that the error message that's returned to the client is very generic and does not reveal any information about CLIENTID and/or the existence of 'joe' and/or the validity of the password.

8. Security Considerations

As this extension provides an additional means of communicating information from a client to a server it is clear there is additional information divulged to the server. This may have privacy considerations depending on the client identity type or its contents. For example, it may reveal a MAC address of the device used to communicate with a server that would not previously have been revealed. While it has been useful to use identifier such as email address for authentication it is easy for these authetication tokens to be shared and/or reused and/or be publically available for other purposes. An IMAP server and or its operators SHOULD not share any CLIENTID information presented with a third party as it may represent or be linked to an individual and SHOULD never be shared in

association with authentication tokens.

As well, while this service extension requires that the identity information only be transmitted over an encrypted channel to reduce the risk of eavesdropping, it does not specify any policies or practices required in the establishment of such a channel, and so it is the responsibility of the client and the server to determine that the communication medium meets their requirements.

9. IANA Considerations

The IANA is requested to add CLIENTID to the "IMAP 4 Capabilities" registry, http://www.iana.org/assignments/imap4-capabilities.

10. References

10.1. Normative References

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[IMAP] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[KEYWORDS] 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>.

[SASL] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[TLS] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

Contributors

Michael Peddemors
LinuxMagic

Authors' Addresses

Deion Yu
LinuxMagic
#405 - 860 Homer St.
Vancouver, British Columbia
CA V6B 2W5

EMail: deiony@linuxmagic.com