Australian Government

Australian Institute of Criminology

Trends & issues in crime and criminal justice

ISSN 0817-8542

No. 512 December 2016

Abstract | The growing use of cloud services in the community means electronic evidence is moving beyond the physical jurisdiction of Australian law enforcement agencies. This paper looks at the legal and technical issues inherent in collecting electronic evidence from online and cloud services and the decline in use of stored passwords for apps in favour of authentication tokens.

The most prevalent authentication token system, OAuth, is described and the utility of these tokens to forensic practitioners is discussed. The legal and technical aspects are considered, with a view to enhancing the capability of law enforcement and practitioners in an increasingly complex legislative and technical environment.

NDLERF
National Drug Law Enforcement Research Fund

Digital forensics in the cloud era: The decline of passwords and the need for legal reform

Ben Martini, Quang Do and Kim-Kwang Raymond Choo

Online and cloud computing services are increasingly prevalent to the point where, for many people, they are integral to communication in their daily lives. From a criminal justice perspective, this makes them key sources of evidence for prosecuting both traditional and online crime (Quick, Martini & Choo 2014). However, the successful prosecution of individuals who commit crimes involving electronic evidence relies upon two major factors. The first is appropriately resourced law enforcement agencies and forensic practitioners who are able to collect, analyse and present the evidence (Quick and Choo 2014); the second is a legislative framework that facilitates the collection of evidence in the modern era, particularly where much of the relevant electronic evidence may be stored beyond the jurisdiction of the investigating law enforcement agency, and the nation's borders generally (Choo 2010, 2014). Given the relatively recent advent and changing face of cloud computing technologies, and their widespread use, it is important to discuss these factors when looking at the challenges of collecting evidence from cloud computing systems in the current statutory environment, and the technical challenges of authentication in forensic collection—particularly as cloud service providers continue to enhance the security of their services.

This paper first discusses the various provisions for search and seizure of evidence available to Australian law enforcement agencies. It then focuses on the increasing emphasis placed on the security of online services in recent times and the effect this has had on authentication. Where digital forensics are used to collect evidence of a crime within a law enforcement agency's physical jurisdiction, it is common practice to take a physical bit-stream image of the storage in the devices to be forensically analysed. The methods used to collect this image do not generally require authentication, as the process requires physical control of the device. In the online environment users—and, generally, forensic practitioners—do not have physical access to the storage devices hosting their data.

There is still a need, however, for a copy of the electronic evidence to be analysed and, ultimately, presented.

It has become relatively common for forensic practitioners, particularly those outside the jurisdiction of the cloud service provider, to obtain copies of electronic evidence using similar technical means as the user (eg a client-visible application programming interface or API). There is some doubt as to the forensic soundness of this process, particularly in terms of the lack of preservation measurements such as cryptographic hashes (which act as a unique identifier) matching for the source and the forensic image (an identical duplicate of the data source), and the increased potential for contamination, depending on the software tools used. As such, the most appropriate means of undertaking such a collection remain undetermined.

Regardless of the tools or methods used, most online services require authentication for every data access request. This relies on the practitioner gaining access to the user's credentials. These credentials have traditionally been a username and password, and law enforcement agencies have developed various methods, such as extracting them from application caches, to obtain them from suspects or their devices. As service providers improve the security of their services, however, there are fewer avenues available for collecting these credentials. This is at least in part due to the increase in token-based authentication systems in contemporary apps. A token-based system typically requires the user's credentials only once, at initial logon. These credentials are then used to obtain one or more tokens that are used for future authentication as required. This is discussed in greater detail in the *Authentication systems* section of this paper.

To ensure law enforcement are able to maintain and, optimally, improve their current capabilities in evidence collection from online and cloud services, they must clearly understand the operation of contemporary authentication systems. This paper assists in reducing the gap in documented knowledge of this area.

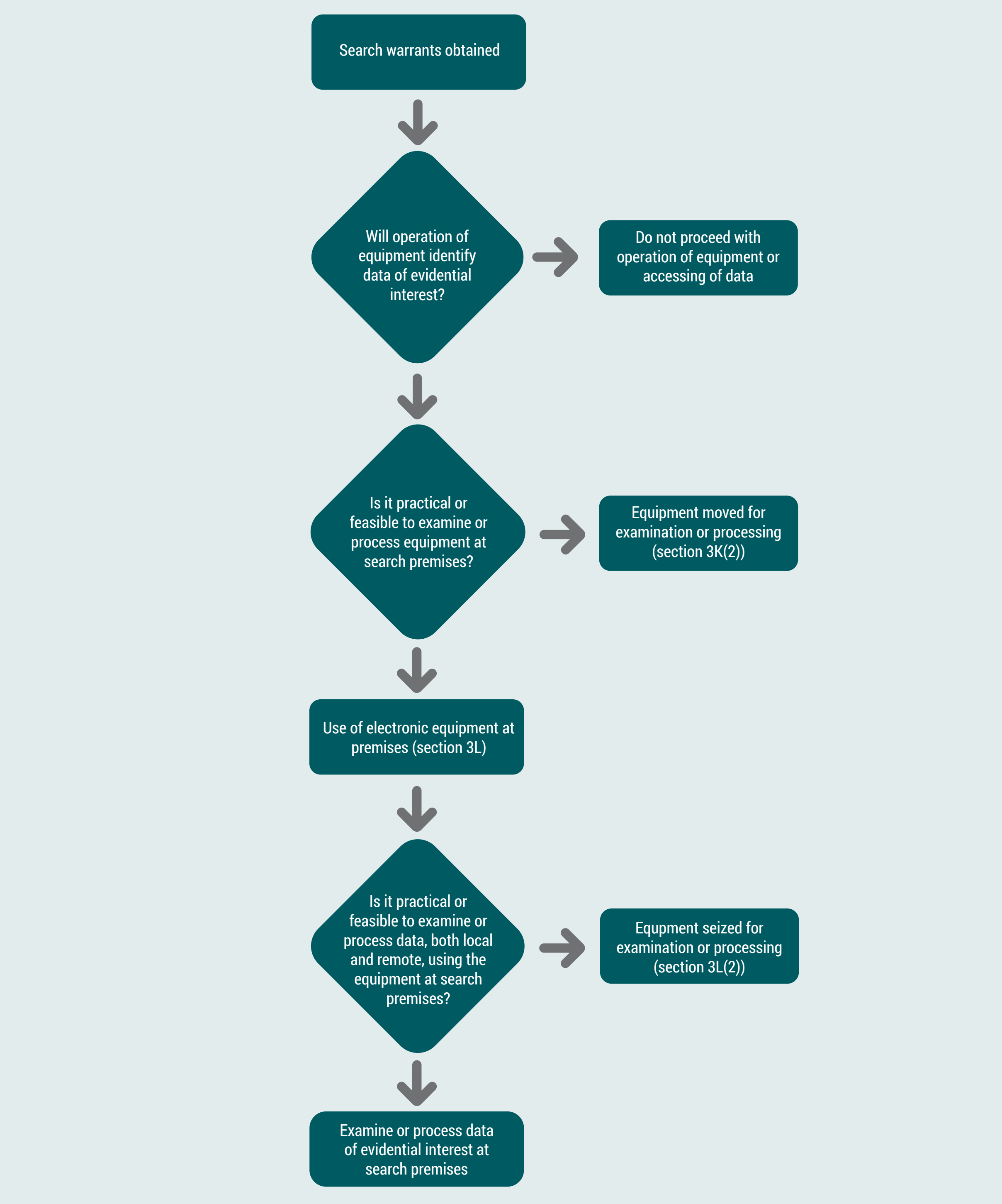
Evidence collection: The Australian legislative perspective

Before data can be forensically analysed, it must be identified and preserved. As noted in an earlier work (Hooper, Martini & Choo 2013), law enforcement agencies can access certain data without the need for a formal warrant—for example, cloud service providers are obliged to render 'reasonable assistance’ under section 313 of the *Telecommunications Act 1997* (Cth) for the purposes of, among other things, enforcing the criminal law and laws imposing pecuniary penalties, assisting to enforce the criminal law of a foreign country, protecting the public revenue and safeguarding national security. It is also possible for Australian law enforcement agencies to apply for a warrant under the *Telecommunications (Interception and Access) Act 1979* (Cth) to intercept telecommunications in real time, which may allow access to data stored on a cloud server.

Search provisions of the *Crimes Act 1914*

Australian law enforcement agencies can also apply for a search warrant requiring service providers to disclose content data they store. The *Crimes Act 1914* (Cth) contains various provisions concerning search and seizure of evidence in federal criminal matters, with specific provisions relating to electronic searches, including when search warrants can be issued (s 3E), what the search warrant authorises (s 3F), the specific powers available to constables executing warrants (s 3J), the use of equipment such as laptops with forensic-imaging programs to examine or process items (s 3K) and the use of electronic equipment at the premises (s 3L). A constable is defined in section 3 as a member or special member of the Australian Federal Police or a member of the police force or police service of a state or territory. In this paper, the terms ‘executing officer’ and ‘constable’ are used interchangeably.

Figure 1: Current Australian search and seizure powers under the Crimes Act 1914 (Cth)


Search warrants obtained
Will operation of equipment identify data of evidential interest?
Do not proceed with operation of equipment or accessing of data
Is it practical or feasible to examine or process equipment at search premises?
Equipment moved for examination or processing (section 3K(2))
Use of electronic equipment at premises (section 3L)
Is it practical or feasible to examine or process data, both local and remote, using the equipment at search premises?
Equipment seized for examination or processing (section 3L(2))
Examine or process data of evidential interest at search premises

When evidence is likely to be found during the execution of a search warrant, the executing officer will state this in the affidavit supporting the warrant application and in its conditions. Other relevant information includes what type of remote data is expected to be found, the name of the company or entity hosting it and where it is hosted.

When a section 3E search warrant is executed the suspect's electronic equipment, such as desktop computers and smart mobile devices, is examined at the search premises to identify and record evidence of historical access to remotely stored data—for example, a cloud storage app on the equipment may indicate that potentially incriminating data is stored in an online cloud storage account. If this is indeed the case, section 3L allows the executing officer to use the electronic equipment to access the remotely stored data, if he or she is satisfied the equipment can do so and that the process for doing so fulfils the conditions of the warrant.

Where assistance is required to operate the equipment or access specific data held on it, the executing officer can apply to a magistrate under section 3LA(1) for an order requiring a specified person to provide reasonable and necessary information or assistance to allow the executing officer to:

(a) access data held in, or accessible from, a computer or data storage device that:

(i) is on warrant premises; or

(ii) has been moved under subsection 3K(2) and is at a place for examination or processing; or

(iii) has been seized under this Division;

(b) copy data held in, or accessible from, a computer, or data storage device, described in paragraph (a) to another data storage device;

(c) convert into documentary form or another form intelligible to an executing officer:

(i) data held in, or accessible from, a computer, or data storage device, described in paragraph (a); or

(ii) data held in a data storage device to which the data was copied as described in paragraph (b); or

(iii) data held in a data storage device removed from warrant premises under subsection 3L(1A) (*Crimes Act 2014*).

Section 3K also permits equipment to be brought to the search premises to examine and process items, including any electronic devices, found there to determine whether they may be seized. As discussed below, it might be possible to search for authentication artefacts, such as tokens, during this process.

In the event it is not feasible or practical to access the data using the equipment located at the search premises (eg due to bandwidth constraints or concerns about the forensic soundness of the equipment), section 3L(2) of the Crimes Act permits the seizure of the equipment and any disk, tape or other associated device. Section 3K(2) allows items to be moved from the search premises to another place for examination for up to 14 days if it is reasonably suspected that the items constitute evidence. An executing officer may apply to an issuing officer for one or more extensions of that time, if he or she reasonably believes the item cannot be examined or processed within 14 days or any time as previously extended (s 3K[3B]).

If something is moved to another place to be examined or processed under section 3K(2) the executing officer must, if it is practical to do so:

(a) inform the person referred to in [later paragraphs] (as the case requires) of the address of the place and the time at which the examination or processing will be carried out; and

(b) allow that person or his or her representative to be present during the examination or processing (*Crimes Act 1914*).

However, the executing officer does not need to comply with these provisions if they believe on reasonable grounds that doing so might:

(a) endanger the safety of a person; or

(b) prejudice an investigation or prosecution (*Crimes Act 1914*).

Sections 3LAA and 3ZQV permit any items seized or moved to be used to access data, including data held elsewhere—for example, in an online cloud storage account—to determine whether data (ie evidence) is stored on or remotely accessible from the electronic equipment, and to obtain access to that data. The equipment can be used either before or after the expiry of the warrant. The legislation explicitly allows only equipment located at, moved or seized from the search premises to be used to access remotely stored data—in other words, a forensic copy of the data or equipment, including those of the investigating law enforcement agency, must not be used to access the remotely stored data.

This restriction presents some issues in terms of forensic soundness best practice. It is commonly recognised that, where practical, the first step in handling an electronic device that may contain evidence is to preserve the evidence. This is most often achieved by taking a forensic image (a bit-for-bit copy) of the device's storage. Further analysis of the evidence is then conducted on the forensic image, rather than on the source. This helps avoid accidental modification of the evidence source and allows more flexible handling of the evidential data—for example, it would allow a practitioner to boot a copy of the seized device in a controlled virtual environment.

Where they are available, it is also preferable to use tools that have been verified as forensically sound, which are designed for use in forensic collection and examination. Such tools ensure evidence—including less obvious elements such as metadata and temporal data—is not inadvertently modified during the collection process. These tools may also use functions such as cryptographic hashes to demonstrate that a true copy of the remote evidence has been made. The only input data generally required by these tools to conduct a collection are the user credentials of the suspect user. These credentials are then used to authenticate access to the online or cloud service.

With this in mind, utilising the client application for the cloud service installed on the seized device—as may be required by sections 3LAA and 3ZQV—is not the most forensically sound option. The best solution would be to take the relevant authentication artefacts from the client device and conduct the collection using law enforcement-controlled, forensically sound hardware and software. However, it is not clear whether law enforcement equipment—that is, equipment brought to the search premises—can be used to process and preserve remotely stored data or whether the executing officer must use the software on the moved or seized equipment to process and preserve the remotely stored data.

Authentication systems

As previously noted, the best way to collect remote cloud data, in terms of forensic soundness, is to use the law enforcement agency's collection environment with collection credentials sourced from client devices. To collect remote cloud data, practitioners must thoroughly understand how the authentication systems used by cloud services work, both to design a collection system and to understand the limitations of the authentication system (eg credential expiry) and its use in common collection operations.

Password-based authentication has been the de facto standard in computer system authentication since the introduction of multi-user computer systems in the 1960s. Initially, developers of online and cloud mobile apps often stored (or cached) password-based user credentials on the device. This allowed the app to access the user's credentials and make continuous requests for up-to-date data without user intervention. This caching was necessary primarily due to the 'always-on' nature of mobile apps, where users expect continuous updates from an online service, often requesting data that is protected by authentication. Recently, however, developers have had to consider other approaches to authentication.

The approach of storing user credentials in plaintext (ie unencrypted) on the device presents a security risk to the user. Malicious parties, or apps they control, could obtain the user's stored credentials. Using these credentials, the malicious party could impersonate the user when communicating with the service.

An example of this is the LinkedIn app for Android which originally stored the user's credentials, including their password, in plaintext on the device (Ante 2011). This made it easy for appropriately resourced malicious parties to access the protected data the service stored for the user. As a result, developers were forced to deviate from authentication schemes that verified the user's username and password each time the app and the service communicated.

Token-based authentication is a popular contemporary authentication system that does not require the user's password-based credentials to be stored on the device. Rather than requiring the user's username and password for each session with a particular service, the user authenticates once with their username and password and receives a token (Satyanarayanan 1990). This token can be time-limited—for example, it may expire after 24 hours. The app can use the token to authenticate the user with the service as long as the token remains valid.

The user is generally required to reauthenticate with their password-based credentials to generate a new valid token upon expiration of the old token, or utilise a 'refresh' token as discussed later in the paper. The advantages of token-based authentication schemes include:

- users authenticate with their credentials only when their tokens expire, providing fewer opportunities for man-in-the-middle attacks to obtain the user's password-based credentials;
- user passwords are typically short in length and can be guessed. Authentication tokens, on the other hand, are often comprised of a long string of characters. This makes it computationally infeasible to perform brute force attacks on services utilising token-based authentication. The reduced frequency with which users need to authenticate using their password also allows developers to harden this authentication interface; and

- tokens can be rescinded by the service at any time for security reasons. Tokens can also be given a subset of the user's privileges—for example, an instant messaging app authenticating with the user's Facebook account to retrieve the user's friends list can be given an access token with limited access to resources. In this case, the app would only be able to access the user's friends list and not all other user data available from Facebook. This process would be impossible, or very difficult to achieve, if the user's password was shared.

Prominent authentication systems

To determine what the most prominent contemporary authentication systems were for online and cloud applications, the study analysed 10 popular Android apps (and their Windows equivalents where available), selected for their popularity and their use of remote authenticated services. The apps analysed are listed in Table 1.

On the Windows platform, official Windows Store apps were selected where available. Where apps were available on Windows but not available via the Windows Store, desktop apps were used. Instagram and LinkedIn were not available on Windows as desktop or Windows Store apps at the time of research. Facebook Messenger functions were integrated into the Windows Store Facebook app.

Android

To determine the prominent online authentication systems used on Android devices, the selected popular apps were forensically analysed. The study utilised the latter examination and analysis stages of the method outlined in Martini, Do and Choo (2015) for forensic analysis of cloud apps on Android. This primarily involved analysing the files stored by the apps on the Android device, examining the account management system and, where necessary, analysing the app's operation and/or code.

Android apps typically store authentication data in two locations: in the app's private storage folder on the device's internal storage, and by using Android's AccountManager API. AccountManager API is considered more secure than internal storage due to the additional layer of hardware-backed protection on credentials stored this way (if available on the device).

The first step in ascertaining what authentication systems are used by the selected apps is determining where each app stores the authentication data (ie the username and password or access tokens). The details of this examination are presented in Table 2.

The study's findings show only three of the 10 apps selected utilised the AccountManager API to store their authentication data. The findings also show all 10 selected apps did not store the user's credentials (ie username and password) directly but rather stored authentication tokens. In addition, two of the 10 selected apps further obfuscated their stored authentication tokens in an attempt to thwart malicious attackers seeking to obtain and use these tokens.

All 10 Android apps appear to use the OAuth authentication protocol, with nine of the apps appearing to use OAuth version 2.0. Dropbox was the only service examined still using version 1.0 of the OAuth protocol. Based on the popularity of the selected apps and the fact that all 10 utilised the OAuth protocol, it is clear that OAuth has become the standard for online authentication of Android apps. The OAuth protocol is discussed in greater detail later in this paper.

Table 1: Apps selected for analysis (at 7 October 2014)

App	Version (Android)	Version (Windows Store)	Category (and Category Rank) in the Google Play Store
Box	3.2.1	2.0.0.12	Business (13th)
Facebook Messenger	10.0.0.17.14	N/A	Communication (1st)
Skype	5.0.0.49715	3.1.0.1005	Communication (2nd)
OneDrive	2.8.1	6.3.9600.16384	Productivity (9th)
Dropbox	2.4.5.10	2.0.0.0	Productivity (10th)
OneNote	15.0.3130.1014	16.0.3030.1024	Productivity (28th)
Facebook	15.0.0.20.16	1.4.0.9	Social (1st)
Instagram	6.4.4	N/A	Social (2nd)
Twitter	5.23.0	1.1.13.8	Social (4th)
LinkedIn	3.4	N/A	Social (6th)

Source: Compiled by Martini, Do and Choo 2015

Table 2: Locations and nature of authentication data on selected Android apps

App	Stored using AccountManager	Stored in internal storage	Token obfuscation	OAuth version
Box	No	Yes	No	2.0
Dropbox	No	Yes	No	1.0
Facebook	No	Yes	No	Derivative of 2.0
Facebook Messenger	No	Yes	No	2.0
Instagram	No	Yes	Yes	2.0
LinkedIn	No	Yes	No	2.0
OneDrive	Yes	No	No	2.0
OneNote	Yes	No	Yes	2.0
Skype	No	Yes	No	2.0
Twitter	Yes	No	No	2.0

Source: Compiled by Martini, Do and Choo 2015

Table 3: Locations and nature of authentication data on selected Windows apps

App	Stored using Credential Manager	Token obfuscation	Authentication system
Box	No	No	OAuth 2.0
Dropbox	Yes	Yes	OAuth 1.0
Facebook	Yes	No	Derivative of OAuth 2.0
OneDrive	Yes	Unknown	Appears to be WS-Security
OneNote	Yes	Unknown	Appears to be WS-Security
Skype	Yes	Unknown	WS-Security
Twitter	Yes	No	OAuth 1.0

Source: Compiled by Martini, Do and Choo 2015

Windows

The study utilised a similar approach on the Windows platform to determine what authentication systems are used by Windows apps. Apps may store their credentials in a number of locations on a desktop computer running the Windows OS. One commonly used location is the app's 'appdata' directory, located within the user's home directory.

Windows does not, by default, enforce suitable protections on folders or files in the appdata directory where credentials are stored. This makes the appdata directory a poor choice for storing user credentials. A more secure method of storing user credentials on Windows, commonly used by Windows store apps, is the Windows Credential Manager.

Credential Manager is the inbuilt credential storage system available in recent versions of Windows. It can securely store information such as passwords, usernames and access tokens. By default, credentials can be stored in Credential Manager as either web credentials or Windows credentials, and are saved to storage in an encrypted file.

One major difference between these types of credential storage in later versions of Windows is that the contents of the web credentials store can be viewed within the Credential Manager if the account password is entered, but a user cannot view the contents of Windows credentials natively. Web credentials are also stored by Internet Explorer when the user has opted to save a set of credentials for a particular website. Apps may store their own credentials as either web or Windows credentials if they wish to utilise the Credential Manager.

To access the contents of the web credentials (without access to the account password) and Windows credentials directories in plaintext, the Credential Manager dynamic-link libraries must be used. These libraries include 'vaultcli.dll', 'vault.dll' and 'advapi32.dll', which can be utilised via API calls in order to read both Web and Windows credentials in plaintext. This was the technique utilised by the study to obtain the stored credentials. However, the user needs to be logged into the device being analysed; practitioners attempting to recover credentials from an offline forensic image will need to use other approaches.

The Windows Credential Manager system should be further analysed to provide forensic practitioners with the knowledge required to extract credentials from offline images and/or systems where the user's password is unknown. Unfortunately, the study found the Credential Manager system to be somewhat lacking, particularly the sandboxing security improvements introduced with Windows 8.

From the data obtained by the study, it was found that the majority of apps on the Windows platform also used OAuth versions 1.0 and 2.0 (see Table 3).

A subset of apps utilised another authentication protocol apparently based upon WS-Security (a Simple Object Access Protocol [SOAP] security extension) token specification. Detailed analysis of this specification is beyond the scope of this paper; however, as new systems commonly implement Representational State Transfer (REST) rather than SOAP, it seems unlikely that WS-Security will be widely deployed beyond its current usage.

The OAuth protocol

Having found that the majority of apps in the selection of popular online and cloud apps utilised the OAuth protocol, OAuth was chosen as a case study for further discussion, with the intention of providing the reader with a high-level overview of how a token-based authentication system operates from the perspective of a forensic practitioner.

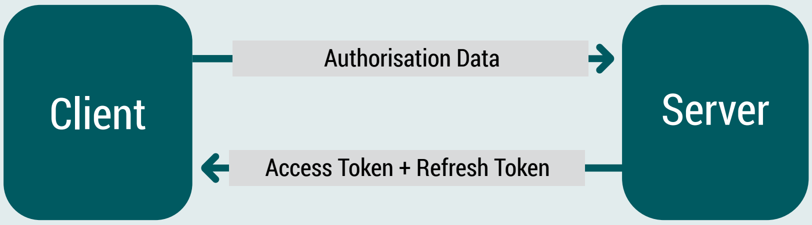
The OAuth protocol is a relatively recent specification. The final draft was released in 2007 and version 1.0 of the protocol was published in 2010 by the Internet Engineering Task Force (Hammer-Lahav 2010). Version 2.0 of the protocol was published in 2012; the second version of the standard was not backwards-compatible with OAuth 1.0 (Hardt 2012). The original OAuth protocol was made obsolete by version 2.0, which has significantly simplified the operation of the token-based authentication system.

One of the most common uses of the OAuth protocol is to provide third-party apps limited access to a user's resources (known as a 'scope'). A third-party app could, for example, request access to a user's Facebook photos. The user would be prompted to authenticate with Facebook's servers directly and allow or deny this request. If the user allows this request, Facebook's servers will return an access token allowing the third-party app limited access to the user's resources (in this case, only the user's photos). As this research focused exclusively on authentication for first-party apps there is little need for an app's servers to return a resource-limited access token—all access tokens stored by these apps should be allowed to access all of a user's resources. The remainder of this discussion is based on this first-party token implementation as explained further later in this paper, and in Figures 2–4.

Initial authorisation

In the initial stages of the OAuth protocol, a client (eg an app) transmits authorisation data to the service's servers. This authorisation data can contain a number of items, known as parameters, and commonly includes the user's username and password and, optionally, a scope (see Figure 2).

Figure 2: Initial authorisation in OAuth 2.0



The scope parameter is used to specify the resources requested and, ultimately, which resources the access token can be used to access. Typically, where a username and password are presented with the initial authorisation data in OAuth 2.0, the access token returned by the server will provide full access to all of the user's resources—because the user's password is generally only provided to first-party client apps, as users may not trust third-party clients.

When using password-based authorisation in the OAuth 2.0 protocol, the parameters must be transmitted using Transport Layer Security (TLS). This is mandatory for secure operation, as the HTTP transmission contains the user's credentials in plaintext. After successfully authenticating with the server the client receives an access token and a refresh token, generally in JavaScript Object Notation (JSON) format. The client may also receive an 'expires_in' parameter denoting the period of time the access token is valid for, along with any other parameters specified by the service provider's implementation.

Accessing a resource

In order to access a protected resource (eg bank account details or private messages on a social networking website), the client is required to transmit the valid access token along with the requested resource's URL on the server. Typically, no further parameters are required in order to obtain the protected resource.

If an access token is invalid (eg the access token has been invalidated or has expired), then the server will inform the client. The client then sends the refresh token obtained from the original password-authentication to the authentication server. If this refresh token is valid (as refresh tokens may have an expiry period), the authentication server then returns a new valid access token, and, optionally, a new valid refresh token. This access token can then be used in a new request to retrieve the protected resource.

Figure 3: Accessing a protected resource with a valid access token in OAuth 2.0

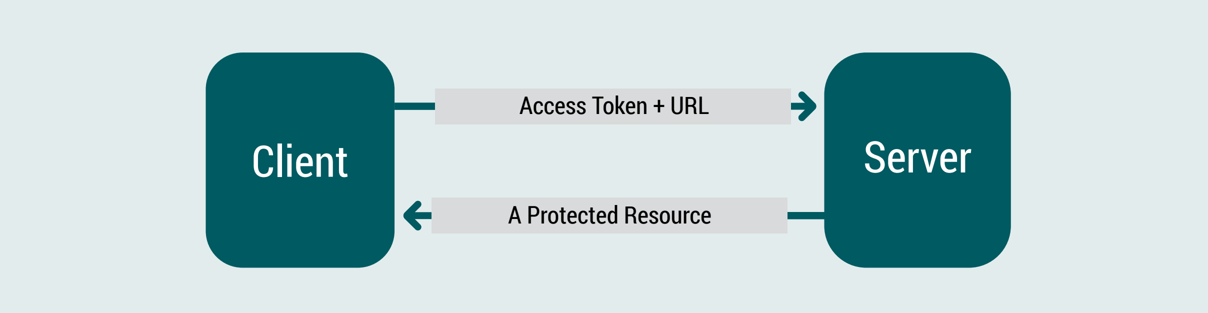
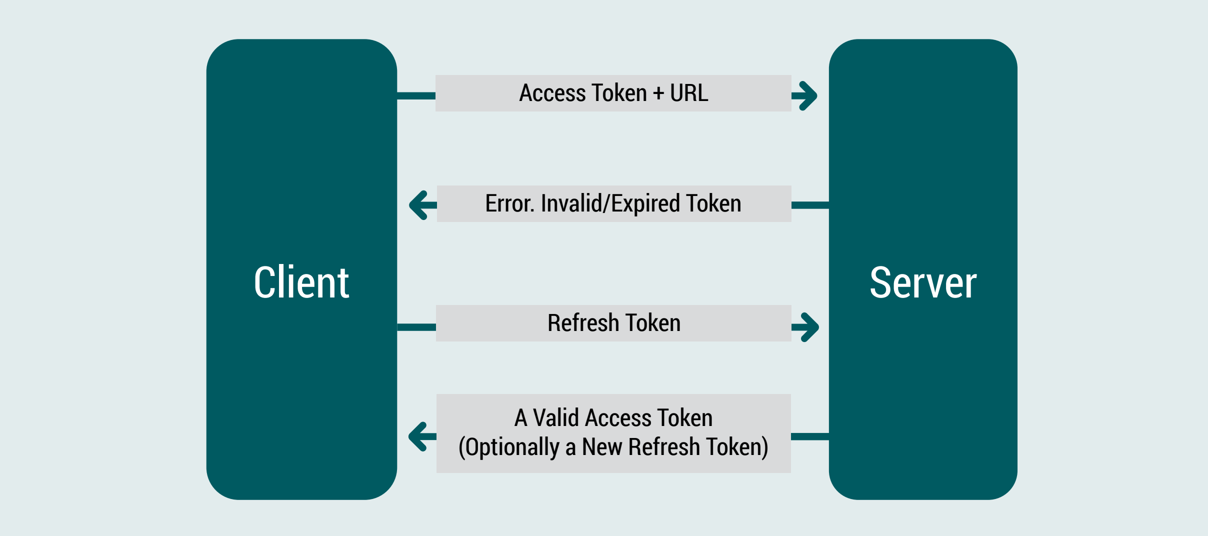


Figure 4: Accessing a protected resource with an invalid access token in OAuth 2.0


Client
Access Token + URL
Error. Invalid/Expired Token
Refresh Token
A Valid Access Token (Optionally a New Refresh Token)
Server

If the service's implementation of the OAuth authentication standard does not utilise a refresh token, the user must re-send their authorisation data.

Generic OAuth authentication process

In summary, the generic process for authenticating a user to an app's online service via OAuth is as follows:

1. the client app prompts the user for their username and password credentials;
2. the app securely sends these credentials and any other necessary data such as the app key or unique ID to the authorisation URL for the service;
3. the authorisation service returns an access token (used to authorise individual resource requests) and a refresh token (used to request new access tokens). The app stores both of these tokens on the client device;
4. when the app needs to make an authorised request for a protected resource, it sends the stored access token as the authorisation. If the access token is valid, the protected resource data is returned and/or the requested operation is executed.

If the access token is invalid or expired, the app will use the stored refresh token with an authorisation URL to request a new access token. The new access token will then be used to request the protected resource, as described in step 4.

The usefulness of this process for forensic practitioners seeking to collect evidence from online and cloud services is clear. The access and/or refresh tokens can be collected from a device and used by the practitioner to access user data, even when usernames and passwords are unknown. The use of access tokens can be embedded in forensic collection applications designed to obtain evidence from online and cloud services in a forensically sound manner.

Another potential advantage of access tokens is they may remain valid even after the user changes their password-based credentials, depending on the service provider's implementation. The access token may also be useful—again, depending on implementation—when access to the user's data is required without their knowledge; the user may continue to use the service meanwhile.

OAuth 1.0 vs OAuth 2.0

The primary difference between versions 1.0 and 2.0 of the OAuth protocol is that OAuth 2.0 relies entirely on HTTPS in order to secure its transmission to and from servers (Hardt 2012). On the other hand, OAuth 1.0 requires the use of cryptography on both the client and the server, in order to secure all authorisation of and access to protected resources (Hammer-Lahav 2010). Both versions of the OAuth protocol require the app's unique secret (as assigned by the authentication provider) upon initial authentication with the user's username and password. However, OAuth 1.0 requires that the app's secret be transmitted with each protected resource request (along with the access token), whereas OAuth 2.0 only requires the access token. In addition, OAuth 1.0 requires the transmission of a nonce (a randomly selected value), the signature method and version numbers during initial authentication given it does not rely on HTTPS for secure transmission.

Another significant difference between the two versions of the protocol is that OAuth 1.0 does not utilise refresh tokens. Any access tokens generated are likely to be valid for an extended period of time—potentially perpetually. The use of refresh tokens in OAuth 2.0 allows scope-limited tokens with short lifespans and new tokens to be obtained without user involvement.

Concluding remarks

To keep pace with the growth and changing face of criminal activity, it is essential that evidential material can be identified and preserved regardless of whether it is held domestically or overseas. A number of governments have sought to enhance their technical capability (and, in some instances, to circumvent or weaken existing security measures) and introduce legislation allowing national security and law enforcement agencies to conduct online surveillance.

As demonstrated in the analysis of popular cloud apps presented in this paper, practitioners must adapt to the use of tokens as the most common means of stored authentication. Practitioners who have seized and analysed client devices such as laptops and smartphones will often be in a position to locate authentication tokens as part of this process.

Forensic practitioners will have to adapt their procedures for examining electronic evidence to ensure they obtain these authentication tokens where available. They will also need to maintain a working knowledge of the authentication token system if they are to be able to exploit the tokens to collect further evidence stored on remote systems.

One major challenge is the time-consuming process of establishing that tokens exist for particular products and platforms, their type and usage. Researchers working in this field can assist practitioners by thoroughly analysing popular software products and platforms to guide practitioners. With this information, cloud-hosted data—an evidence source of significant and growing importance—will be accessible to forensic practitioners in most cases and evidence hosted in the online environment will, at least technically, be available for presentation to court.

It is, however, unclear whether existing Australian law permits the real-time use of remote evidence preservation and collection processes and tools to preserve evidence stored or held overseas without a mutual assistance request.

If this direct-collection approach is deemed infeasible, Australian law enforcement agencies may also have access to alternative evidence collection approaches. For example, agencies can issue a domestic preservation notice to a carrier requiring the carrier to preserve all stored communications they hold which relate to the person or telecommunications service specified in the notice under section 107G of the *Telecommunications (Interception and Access) Act 1979* (Cth). Domestic preservation notices cover any stored communications that might relate to either the contravention of certain Australian laws or security; see Division 2 of the *Telecommunications (Interception and Access) Act 1979* (Cth).

Where evidence is determined to be hosted or stored outside Australia, Australian law enforcement agencies make a mutual assistance request via the federal Attorney-General to obtain the evidence in a form suitable for admission in Australian court proceedings under the *Foreign Evidence Act 1994* (Cth); see the Australian Government Attorney-General's Department (2013). Mutual assistance is a reciprocal process. Where a foreign nation has requested the preservation of evidence held in Australia only the Australian Federal Police can serve the foreign preservation notice on an Australian carrier, and only if the foreign country requested the preservation in accordance with section 107P of the *Telecommunications (Interception and Access) Act 1979* (Cth). Foreign preservation notices cover stored communications that might relate to the contravention of certain foreign laws; see Division 3 of the *Telecommunications (Interception and Access) Act 1979* (Cth). Access to the information is then regulated by the *Mutual Assistance in Criminal Matters Act 1987* (Cth) which, in principle, affords sufficient safeguards to ensure access is consistent with Australian values.

Whether accessing evidence stored or held remotely (eg in overseas cloud storage accounts) could result in the violation of a foreign law is a grey area. It is therefore important that digital forensic researchers collaborate with legal and policy scholars and practitioners to ensure the legal effectiveness of any remote evidence preservation and collection processes.

From a technical perspective, work on the remote collection of evidence should continue. Future work should include an in-depth analysis of some of the less prominent authentication systems in use, and a discussion of the most appropriate ways of exploiting their functionality for the purposes of digital forensics.

References

URLs correct as at February 2016

Ante SE 2011. Some Top Apps Put Data at Risk. *Wall Street Journal* 8 June. http://blogs.wsj.com/digits/2011/06/08/some-top-apps-put-data-at-risk/

Attorney-General's Department 2013. *Fact sheet—Mutual assistance overview*. https://www.ag.gov.au/Internationalrelations/Internationalcrimecooperationarrangements/MutualAssistance/Pages/default.aspx

Hammer-Lahav E 2010. *The OAuth 1.0 Protocol.* The Internet Engineering Task Force. http://tools.ietf.org/html/rfc5849

Hardt D 2012. *The OAuth 2.0 Authorization Framework.* The Internet Engineering Task Force. http://tools.ietf.org/html/rfc6749

Hooper C, Martini B & Choo KKR 2013. Cloud computing and its implications for cybercrime investigations in Australia. *Computer Law & Security Review* 29(2): 152–163

Martini B & Choo KKR 2014. Remote Programmatic vCloud Forensics: A Six-Step Collection Process and a Proof of Concept. In *Proceedings of 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).* Piscataway, NJ: IEEE Computer Society Press: 935–942

Martini B, Do Q & Choo KKR 2015. Conceptual evidence collection and analysis methodology for Android devices, in Ko R & Choo KKR (eds), *Cloud Security Ecosystem.* Waltham, MA: Syngress, an imprint of Elsevier: 285–307

Quick D, Martini B & Choo KKR 2014. *Cloud storage forensics.* Waltham, MA: Syngress, an Imprint of Elsevier

Satyanarayanan M 1990. Scalable, secure, and highly available distributed file access. *IEEE Computer* 23(5): 9–18

Kim-Kwang Raymond Choo is a Research Professor with the University of Texas at San Antonia and the University of South Australia. Ben Martini is a Research Fellow at the University of South Australia. Quang Do is a PhD scholar at the University of South Australia.

General editor, *Trends & issues in crime and criminal justice* series: Dr Rick Brown, Deputy Director, Australian Institute of Criminology. Note: *Trends & issues in crime and criminal justice* papers are peer reviewed. For a complete list and the full text of the papers in the *Trends & issues in crime and criminal justice* series, visit the AIC website at: aic.gov.au

ISSN 0817-8542

GPO Box 1936
Canberra ACT 2601, Australia
Tel: 02 6268 7166

Disclaimer: This research paper does not necessarily reflect the policy position of the Australian Government