Identity Management Landscape Overview

Purpose

The purpose of this document is to provide technical information overview from leading industry sources for the data privacy protection in an environment of accelerating criminal activity for data theft.

Among other things, this report provides an overview of the Identity Management landscape.

Standards take time to develop, and as the rate of malicious attacks has outpaced standards bodies knowledge and resources, this document surveys the leading industry sources for information about best practice and current leading knowledge bases for circumvention and prevention of exploitation of vulnerabilities in current web technologies.

As the functionality of Cloud Information Services increases, and the level of interest in gaining access to private data for high net worth individuals and public figures, it is advisable to ensure encryption of all communications from vehicles is best practice.

Identity Management Technology Considerations

OAuth 2.0

Standard Resource Owner Password Style OAuth 2.0 Implementation

The abstract flow for the OAuth 2.0 protocol is

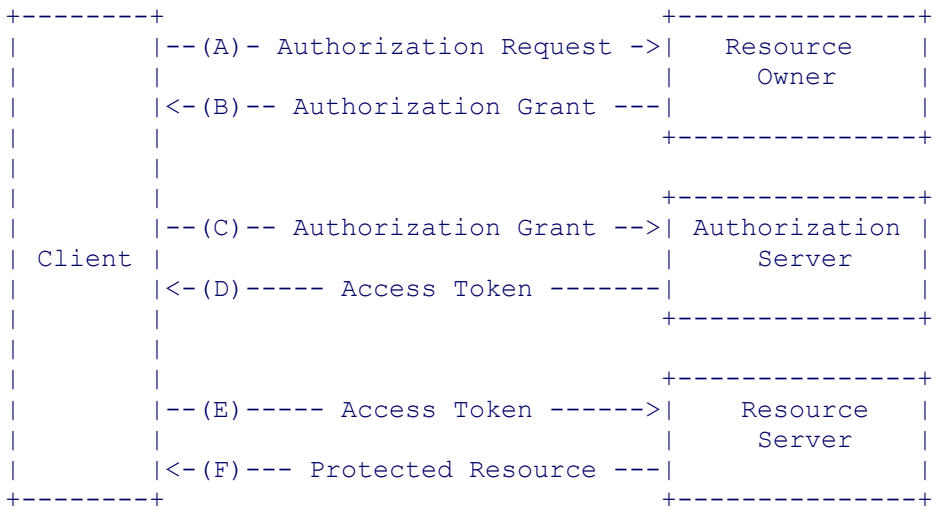

FIGURE 1: ABSTRACT OAUTH2.0 PROTOCOL

There are a number of common scenarios involved in the implementation of OAuth 2.0.The four major types are:

1. Authorization Code – The client application asks for a code provided by a third party, such as Google, Facebook, Twitter.
2. Implicit
3. Resource Owner Password Credentials
4. Client Credentials

The Resource Owner's Password Credentials Flow scenario is used frequently, though really only recommended for use to integrate legacy systems when no other method is available. However it is commonly used as it is relatively to simple to deploy. .

The standard OAuth 2.0 steps involved in this system user authorisation scenario are identified as:

- The System User(Resource Owner) authorises the Client Application (Client) to access content (resources) that are accessible through the application, allowing the Client Application to transmit their UserID and Password to the OAuth 2.0 Server (Authorisation Service)
- The Client Application passes this request to the OAuth Server
- The OAuth 2.0 Server returns an Authorisation Code to the Client Application.
- The Client Application returns the Authorisation Code plus Shared Secret to the OAuth Server

- The OAuth 2.0 Server returns a Token to the Client Application
- The Client Application provides the Token with any request for content to the Client Application Server (Resource Server)

Roles:

1. System User is the Resource Owner
2. Client Application is the Client.
3. OAuth 2.0 Server is the Authorisation Server
4. Client Application Server is the Resource Server

The authorisation/authentication process established as the System User is associated with a service involves the supply of an OAuth 2.0 Resource Owner Password Credentials scenario delivered by the Oauth 2.0 Service, based on UserID and Password collected from the System User as he/she first accesses a Resource Content subscription.

There are a number of other standard OAuth 2.0 scenarios, which, when implemented correctly, provide stronger authentication and authorisation capabilities. They are:

SSL Using PKI

Protecting enterprise endpoints is still the critical factor in ensuring information security, in transit, as well as on premises or hosted in the cloud.

Endpoint protection often encompasses

- Full-disk and file encryption
- Endpoint data loss prevention
- Vulnerability assessment
- Application controls
- Mobile device management (MDM)

Many SSL deployments use symmetric encryption, without real visibility of the policy relating to certificate validity/revocation checking etc.

An improvement may be to provide some policy advice on certificates, and also to establish an initial SSL handshake protocol with asymmetric encryption (PKI) to improve the transport of system user credentials.

Industry Identity Management Landscape

Single Sign On is easy to achieve, however the vulnerabilities for personal data have been growing exponentially over the past two years, as cloud hosted mobile applications proliferate. An SSO that secures personal data and credentials is somewhat more difficult.

To facilitate the delivery of content from the growing number of service providers, identity management is moving towards accommodating the current and emerging standards for mobile devices.

Successful attack on data and functionality for financial gain has experienced exponential growth in the last two years, according to reports from a number of highly respected sources, such as the Cloud Security Alliance, Cloud Council, IBM and others.

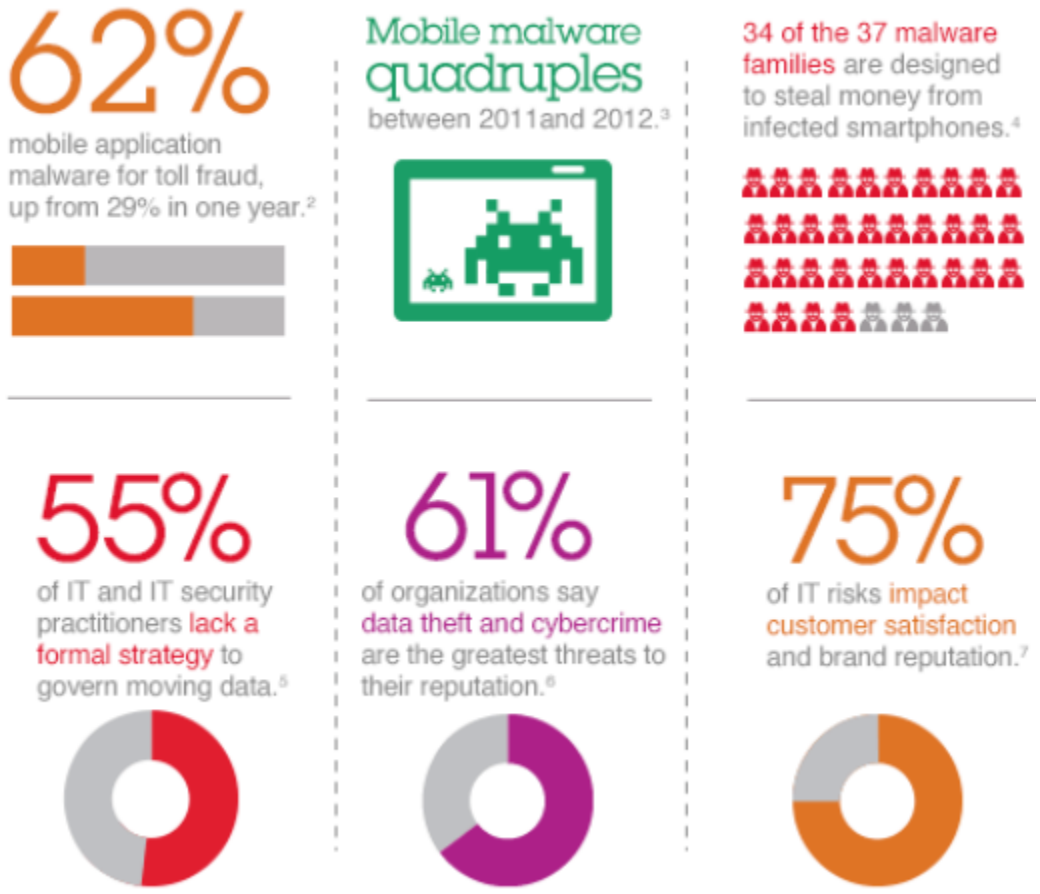

FIGURE 2: IBM MOBILE SECURITY SURVEY 2013

Standards and Best Practice

The cloud model aligns closely with mobile applications in terms of addressing scale, performance, availability and access. The challenge is the race against the clock to fix the vulnerabilities in delivering mobility from the cloud, in the face of well-funded well-organised efforts to hijack credentials and information that can be used for illegal financial gain.

The ICT industry has proven capable in the past of collaboration to develop solutions to common problems, and this kind of collaborative effort is being mobilised now to defeat the threats posed by organised crime 'hacktivism'.

It is therefore advisable to closely follow the initiatives taken by large ICT players such as Google, Microsoft, IBM, Amazon and Apple in order to take advantage of strategic preventive measures to counterfoil attacks that would pose a threat to customers, and therefore an organisation's reputation.

Taking an integrated approach to identity security means closely examining and evaluating the risks involved in application delivery, not only to onboard computers, but also mobile devices and websites using identity credentials connected to the vehicle.

Proving identity is likely to be targeted for improvement, as the ubiquitousness of mobility and mobile apps appears to be on an unstoppable growth path.

Authentication and authorisation processes are likely to be deliberately obfuscated in execution to foil potential identity hijacks.

OAuth 2.0

OAuth 2.0 is likely to emerge as a standard that is actively developed to counteract the sophisticated attacks on encrypted and secured internet communication channels.

The OAuth 2.0 access authentication and authorisation standard provides for a very specific set of communications between parties to access information.

There are four common scenarios included in the documentation, the least secure of which, designed to integrate legacy systems, is the Resource Owner Password Flow scenario.

While the OAuth 2 standard is flexible enough to allow for hybrid variations of the main scenarios, the design principles have to be respected to ensure security from external threats and attacks.

Single Sign-On

As standards emerge for scenarios of accessing data and content in multiple clouds, through multiple provider services, one emerging protocol seems likely to become widely adopted.

Simple Cloud Identity Management

The Simple Cloud Identity Management (SCIM) protocol defines a simple, RESTful protocol for identity account management operations. SCIM's model is based on the experience of existing schemas and SaaS deployments, placing specific emphasis on simplicity of development and integration, and wherever possible, applying existing authentication, authorisation, and privacy mechanisms.

SCIM recommends using the OAuth 2.0 protocol for SCIM API Call Authentication.

SCIM represents users and groups in JSON and XML schemas, and allows for bindings to other protocols.

In addition to OAuth 2.0, using a service to accept other standard SSO approaches extends the range and the security of customer content.
It may be useful to provide the ability to support tokens from the following technologies

- LDAP
- SAML 2.0
- WS-Federation
- OpenID Connect
- SCIM
- ADFS2

Access control & security policy

An Identity Management Service provider could utilise eXtensible Access Control Markup Language (XACML) Version 3.0

Digital Certificates

Digital Certificates are used to electronically to prove identity. Used to verify the right to exchange encrypted messages based on public and private electronic keys

Current standard digital certificates used for securing links to cloud service providers are PKCS, X.509, and OpenPGP.

The following provides an overview of the components involved in the identity and access management of system users to onboard content services.

SSL

Current implementations of SSL are more than a transport layer security protocol. Standards like OpenSSL are capable of message digests, encryption and decryption of files, digital certificates, digital signatures, and random number generation.

- Most current browsers support Extended Validation SSL. EVL comprises
- Enhanced validation processes for public Certificate Authorities
- More frequent validation of web service providers information (annually)
- Minimum standard 128 bit encryption

The steps involved in a secure SSL handshake between the client and the server:

- Agree on the version of the SSL protocol to use.
- Select cryptographic algorithms.
- Authenticate each other by exchanging and validating digital certificates.

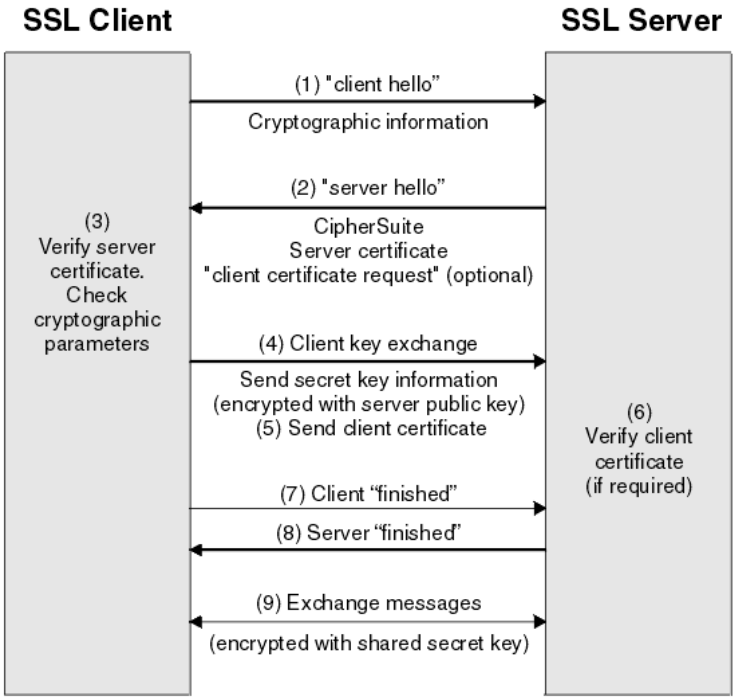

FIGURE 3: OVERVIEW OF THE SSL HANDSHAKE. IMAGE CREDIT IBM

Encryption

Encryption in execution is not a fail safe method of ensuring privacy of communications. There are well known points of weakness. There are also known vulnerabilities for hashing that can be exploited. MD5 is considered by US Homeland Security to be 'broken'. Other vulnerabilities have emerged.

The following material is quoted from the PacketLife website security blog stretch, as a reasonably accurate overview of encryption styles.

Symmetric Encryption

Symmetric encryption may also be referred to as ***shared key*** or ***shared secret*** encryption. In symmetric encryption, a single key is used both to encrypt and decrypt traffic.

Common symmetric encryption algorithms include DES, 3DES, AES, and RC4. 3DES and AES are commonly used in IPsec and other types of VPNs. RC4 have seen wide deployment on wireless networks as the base encryption used by WEP and WPA version 1.

Symmetric encryption algorithms can be extremely fast, and their relatively low complexity allows for easy implementation in hardware. However, they require that all hosts participating in the encryption have already been configured with the secret key through some external means.

Asymmetric Encryption

Asymmetric encryption is also known as Public Key Cryptography (PKI). Asymmetric encryption differs from symmetric encryption primarily in that two keys are used: one for encryption and one for decryption. The most common asymmetric encryption algorithm is RSA.

Compared to symmetric encryption, asymmetric encryption imposes a high computational burden, and tends to be much slower. Thus, it isn't typically employed to protect payload data. Instead, its major strength is its ability to establish a secure channel over a non-secure medium (for example, the Internet). This is accomplished by the exchange of public keys, which can only be used to encrypt data. The complementary private key, which is never shared, is used to decrypt.

Robust encryption solutions such as IPsec implement the strengths of both symmetric and asymmetric encryption. First, two endpoints exchange public keys, which allows for the setup of a slow but secure channel. Then the two hosts decide on and exchange shared symmetric encryption keys to construct much faster symmetric encryption channels for data.

Hashing

Finally, hashing is a form of cryptographic security which differs from encryption. Whereas encryption is a two-step process used to first encrypt and then decrypt a message, hashing condenses a message into an irreversible fixed-length value, or ***hash***. Two of the most common hashing algorithms seen in networking are MD5 and SHA-1.

Hashing is used only to verify data; the original message cannot be retrieved from a hash. When used to authenticate secure communications, a hash is typically the result of the original message plus a secret key. Hashing algorithms are also commonly used without a secret key simply for error checking. You can use the *md5sum* and *sha1sum* utilities on a Linux or Unix machine to experiment with hashing.

IPSec and SSL VPN Overview

IPSec VPN

Although both IPsec and SSL can provide secure access to network applications, they operate differently.

IPsec operates at the network layer of the OSI network model, to establish a VPN tunnel.

It is typically used in conjunction with IKE (Internet Key Exchange) for key management. Together, IPsec and IKE are described in a series of Internet Engineering Task Force (IETF) standards: RFC 2401 to 2411. IPsec supports multiple encryption algorithms (AES, DES, 3DES, RC4) and multiple integrity mechanisms (MD5, SHA-1), as well as authentication via X.509 certificates.

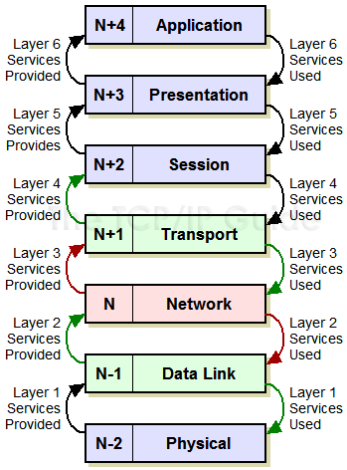

FIGURE 4: OSI NETWORK LAYER MODEL

This can be a positive for multiple applications, but it can become vulnerability if the remote-access client has been compromised.

IPsec is well suited for site-to-site VPNs, because it can be implemented in network devices without any client operating systems or applications having to be modified.

But the necessary deployment of software on client devices can be costly, depending on how many have to be supported.

IPsec VPN networks are capable of performing well in a high-volume environment (1-Gbps traffic with support for thousands of concurrent users) and support fail-over without dropping sessions.

IPsec connectivity can be impaired by firewalls, routers, and proxy devices that reside between the client and the concentrator. In addition, if IPsec VPN sessions are not terminated in the DMZ of a properly managed firewall, in effect it is a hole through their network security providing remote access to the network.

SSL VPN

Familiar SSL is the authentication and encryption mechanism for web transactions. SSL runs on layer 4 (the transport layer) of the OSI model, above TCP/IP and below HTTP.

When a client establishes an SSL-connection handshake with a server

- Server is authenticated to the client, verifying that a server's certificate and public ID are valid and have been issued by a trusted certificate authority.
- Client and server negotiate and select cryptographic algorithms that they both support.
- Client may then be authenticated to the server, and an encrypted SSL connection can be established.

SSL operates transparently across proxies and routers performing Network Address Translation, and it uses TCP ports that are usually left open on firewalls. One potential drawback is that SSL is computationally heavy for both the client device and the SSL VPN device; unless implemented properly, it may require multiple

handshakes per session, thus increasing the computational load. This calls into question the ability of SSL devices to scale to support thousands of concurrent remote users.

There are known and unknown vulnerabilities in SSL, so it is not a complete solution to secure communications. APIs are subject to increased risk from man-in-the-middle attacks even when SSL is used. If an API does not properly use SSL/TLS, all requests and responses between a client and the API server can potentially be compromised. Transmissions of identifiable data can be altered or replayed with improperly configured SSL, or certificates not checked for revocation.

As mentioned in the paragraph above, SSL is a computationally intensive application, from the initial handshake to the encryption of each packet. Factors like the length of each user session, SSL ID, and key reuse (which requires less processing during SSL handshake) all affect general SSL appliance performance. Whether or not the SSL VPN device off-loads SSL handshake and encryption to specially designed hardware will also significantly affect performance. Several network equipment manufacturers make SSL off-loading, load balancing, and TCP/HTTP session management devices.

Mobile Device Vulnerabilities

The following quote is taken from a publication by IBM: Securing the Mobile Network

"By 2015, some 40 percent of enterprise devices are expected to be mobile. Half of mobile applications transmit personal details or device information.

Most mobile platforms are not natively designed to provide comprehensive security, and with the explosive growth in numbers of mobile devices, hackers have a strong incentive to develop new techniques or create attacks aimed specifically at these devices ...vulnerabilities including

- *Credentials that enable access to business or personal accounts*
- *Sensitive data such as confidential business or personal information*
- *Device communication services*
- *The mobile device itself, which can be a jumping-off point to accessing other customer data & credentials"*

At the top of the threat list are lost and stolen devices, but rogue applications, social engineering, malware, identity theft, stolen data, malicious websites and denial of service are becoming more sophisticated and are constantly on the increase "

Scale of deployment also has to be kept in mind. What may start out as a relatively small data transfer may require a different architecture for a large scale adoption of mobile device content delivery. It is advisable to at least keep in mind that a wave of advances for vehicles are heading for large increases mobile device content delivery, over the next 5 – 10 years.

Strategic Risk Assessment

Fhe following categories focus on risk categories and provide information to be used as a basis to prevent common mobile device vulnerabilities to identity and access management

1. Device – Provisioning and Configuration. the organization already has a set of processes using OMA-DM and FUMO nodes for updates. The security of these processes depends on the security of the SSL/TLS over which the updates are made. Attention can also be given to locking/wiping data when vehicle is stolen. (This encompasses being able to securely transfer system user identity and profile data when a vehicle is sold). This may involve being able to identify the head unit itself, as well as the vehicle to which it is attached (preventing rogue device access). The Linux OS also has to be protected from 'Jailbreak/Root' attacks (see note below).
2. Content – restricting access to the personal data stored on the device. Restricting access to any other data, e.g. vehicle information.
3. Applications – As well as using an SDK framework, an app wrapping policy based can be applied on an application basis, depending on risk profile. Where feasible, validation of content APIs is also advisable to enhance security of access to the system. This includes runtime risk detection through a rules based approach as well as whitelisting/blacklisting.. Security technologies can be designed into the applications. Scanning for vulnerabilities during and after deployment is a valuable use of testing resources. Obfuscation makes it difficult for hackers to identify open doors to insert malware, particularly critical given the proximity of APIs to TCU and CAN.
4. Transactions – the information transmitted to and from the device, can be better authenticated with a risk management strategy towards auditing System User access patterns.

Note: Jailbreak/Root (finding access to file system outside the Application Framework) Access Technical Risks for Linux O/S:

1. Some jailbreaking methods leave SSH enabled with a well known default password (i.e. alpine) that attackers can use for Command & Control.
2. Entire file system of a rooted or jailbroken device is vulnerable to a malicious user inserting or extracting files. This vulnerability is exploited by many malware programs, including Droid Kung Fu, Droid Dream and Ikee.
3. Credentials to sensitive applications, such as banking or corporate applications, can be stolen using key logging, sniffing or other malicious software and then transmitted via the internet connection.

User Experience

More attention is placed on continuity of user experience with mobile devices, and mobile device computing is no different.

The Single Sign On user experience can function as a useful launching point for discovering and anticipating identity and access threats.

It is important to try to prevent attacks targeted at both individual customers, and also at the organization itself.

Mobile Aware Identity Management

Secure access management, for authentication of end users, and authorisation to appropriate content and data is a primary focus, and using best practice techniques is essential, given the efforts being made to exploit any weakness in identity management.

- Strong session management is a preventive measure for Man-in-the-Middle attacks.
- Specific application measures to ensure security of applications is advisable, to ensure best practice validation of User and Vehicle access to infotainment systems..
- Best practice is to use software that is context aware, that is, there are rules to determine when to re-authenticate, not only for user events such as change of UserID or Password, but also to address higher risk scenarios, such as insecure content.
- Securing data with rules led by policy as to encryption and management of API endpoints and stored data is best accomplished with software with a proven track record in mobile security.

Software is already available to scan applications for vulnerabilities, and it is advisable to make this a part of testing for initial deployments, to provide input for where to put effort and resources for the longer term.

Best-of-breed mobile security for Identity Managment, has been developed over at least decades to achieve the time and case study hardened capabilities required in an environment where mobile hacking is the province of organised crime engaging in fraud for profit. There is no substitute for in depth knowledge, professional peer groups and standards, libraries of knowledge and experience, and current industry wisdom. This cannot be manufactured or duplicated overnight.

It is highly inadvisable to provide either an ad hoc or bespoke identity management solution. The risks entailed in learning by experience may lead to a loss of data, and should it be traced to the organization, a loss of reputation.

Cloud Vulnerabilities

“With potentially sensitive information passing through the hands of a third party, and with the highly fluid nature of multi-tenant environments, it can be difficult to know exactly where and how well secured your data is at all times.” Cisco /Intel

“In the following figure, it can be observed that the top three cloud providers, Amazon, Google and Microsoft, account for about 56% of all non-transparent incidents of cloud vulnerability. Beginning in 2010, cloud providers became more transparent with their reports of cloud vulnerability incidents, most likely because Amazon became more open about the causes of their incidents “ Cloud Vulnerabilities Working GroupCloud Computing Vulnerability Incidents: A Statistical Overview August 23, 2012; Revised March 13, 2013

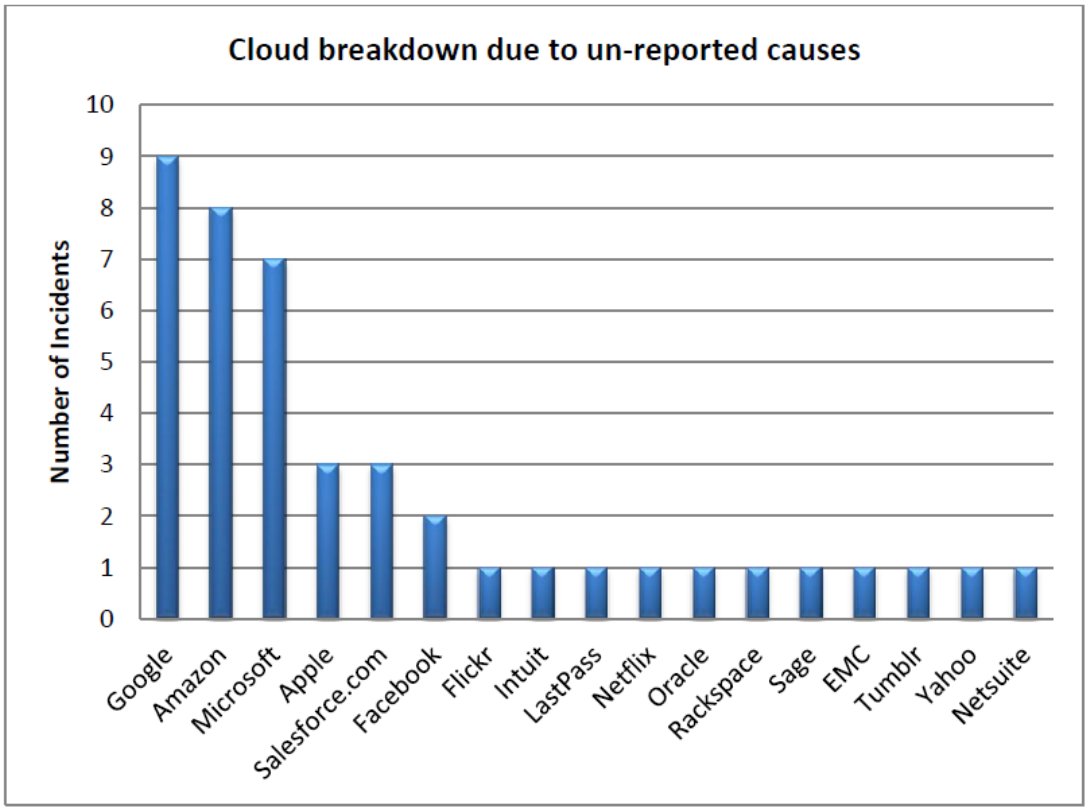

TABLE 1: 2008 - 2011 SECURITY BREACHES BY PROVIDER

The frequency of cloud vulnerabilities increased dramatically over the period 2008 to 2011 in line with the increase in cloud computing services.

Public cloud computing involves a transfer of responsibility and control to the cloud provider over information as well as system components, with no direct control over the management of operations and also a loss of influence over decisions made about the computing environment.

Public clouds deliver scalable services that provide computing power for multiple tenants. Virtualisation can lead to multiple tenant data co-residing in the same hardware, including CPU, memory and storage.

Hypervisors are software or firmware components that are able to virtualize system resources. Even with a virtualisation hypervisor to mediate access between guest operating systems and physical resources, there is concern that attackers can gain unauthorized access to, and control of underlying platforms with software-only isolation mechanisms.

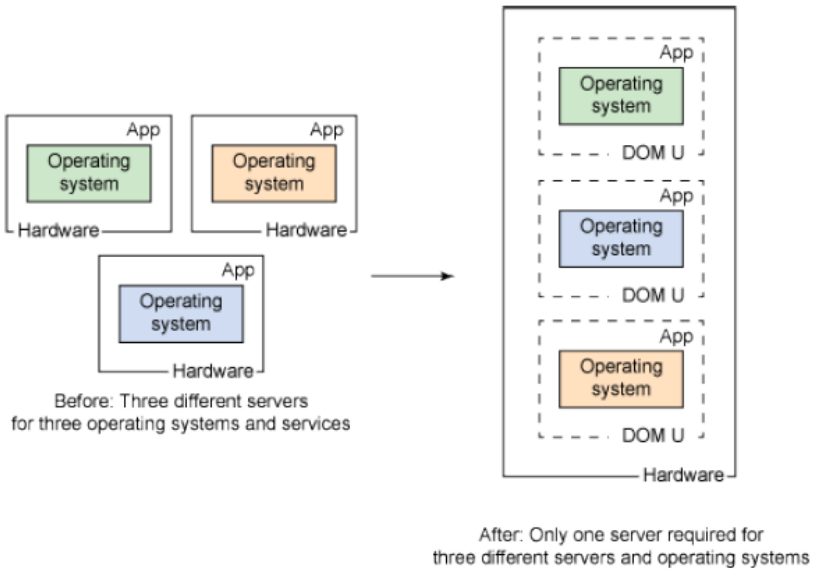

TABLE 2: SIMPLE EXAMPLE OF VIRTUALISATION – IMAGE CREDIT IBM

There are two types of hypervisor

1. Software running on the Operating System
2. Firmware running on the Hardware

Protecting data can be a headache because of the number of ways it can be compromised. Customer data can be maliciously deleted, altered, or unlinked from its larger context.

Hijacking of Accounts or Services: Attacks such as phishing and fraud continue to be an ongoing threat. With stolen credentials, hackers can access critical areas of the the organization cloud presence and potentially eavesdrop on transactions, manipulate or falsify data, redirect customers to illegitimate sites.

Strong identity and access management, including two-factor authentication where possible, strong password requirements, and proactive monitoring for rogue access activity by specifying audit capabilities to service providers is recommended to discourage unauthorised behaviours.

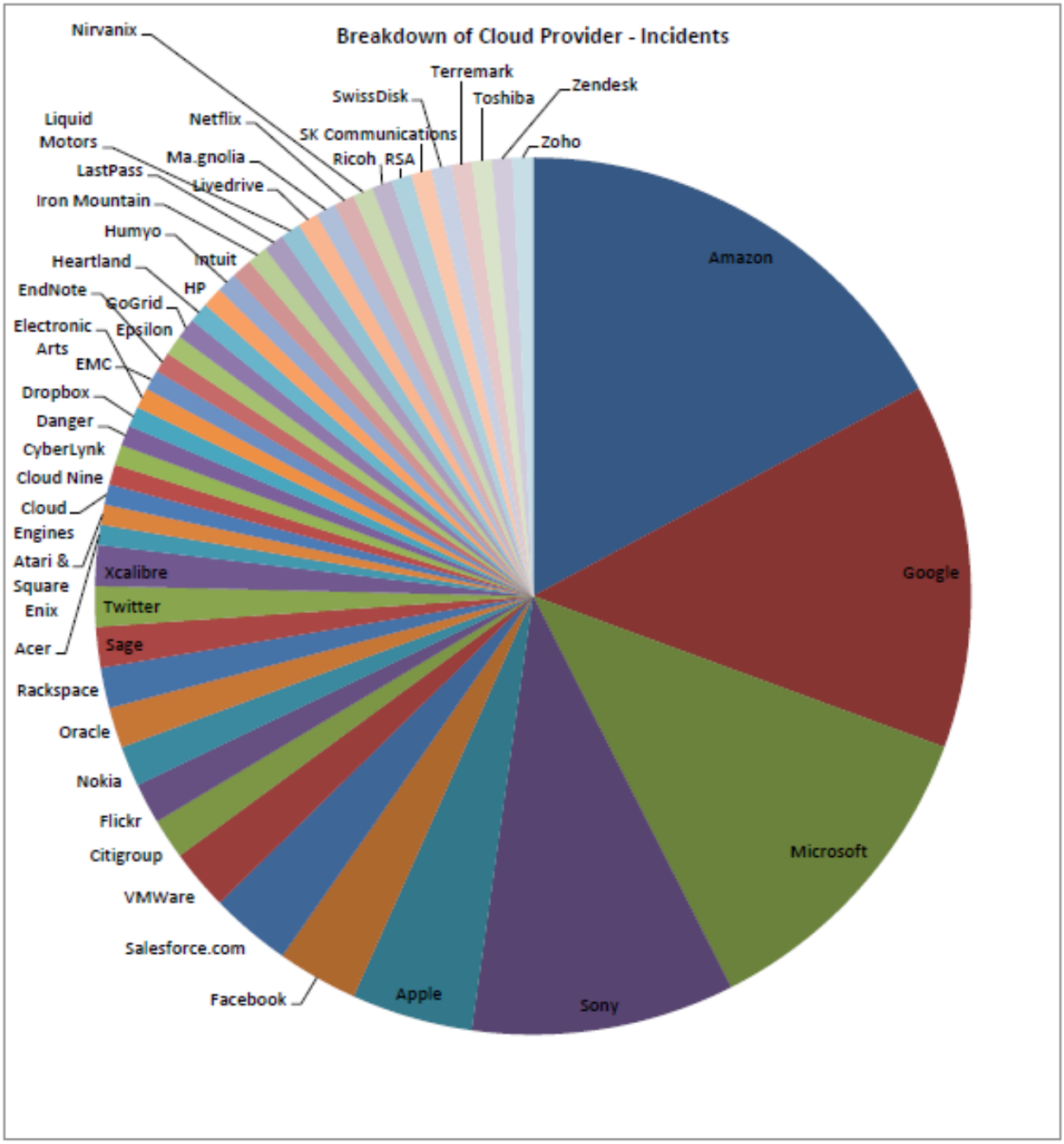

TABLE 3: CLOUD SECURITY ALLIANCE - BREAKDOWN OF INCIDENTS BY PROVIDER

Role Based Access Monitoring

Providing role based authentication, and engaging cloud service that provide not only extensive auditing and access logs on a user basis, but also real-time access for the organization to this information is a strong tool for monitoring access on a per identity basis.

Security Policy

Access to CI/CGI is to be secured by a policy security level that is to be defined for all user accesses to profiles, settings and content.

The following is an industry standard view of policy settings for Identity and Access Management.

Security Level			
Security Measure	Low	Medium	High
Password length (characters)	6+	8+	8+
Password is case-sensitive	No	Yes	Yes
Password can include sequential or repeating characters (like '123456' or 'aaaaa')	Yes	No	No
Passwords must include non-alphabetical character	No	No	Yes
Block user account after N unsuccessful login attempts	Never	10	5
Duration of block	N/A	30 minutes	60 minutes
Minutes of inactivity before expiring user session	240 (4 hours)	240 (4 hours)	240 (4 hours)
Minutes of usage before forcing user to re-login	480 (8 hours)	480 (8 hours)	480 (8 hours)
Minutes to wait before expiring record lock	240 (4 hours)	60 (1 hour)	30 (1/2 hour)
Encryption Level	none	SSL	SSL over VPN
API data validation	no	yes	yes

TABLE 4: EXAMPLE OF ACCESS MANAGEMENT POLICY. SOURCE: PROGRESS SOFTWARE

Recommended Cipher Suites and TLS/SSL

Cryptology is a changing landscape as the mathematics of encryption and decryption become better understood. Currently, improvements in decryption are accelerating at a rate that is faster than encryption advances.

Best Practice

The recommended cipher suites have an algorithm that uses a minimum 128 bit key for encryption of message.

NSA Suite B Cryptography (IETF RFC 6460) required components are

1. Advanced Encryption Standard (AES) developed by US agency NIST in 2001
2. Elliptic Curve Digital Signature Algorithm (ECDSA) digital signatures
3. Elliptic Curve Diffie-Hellman (ECDH) key agreement
4. Secure Hash Algorithm 2 (SHA-256 and SHA-384) message digest

TLS/SSL best practice is the use of libraries that support SSL 3.0 and TLS 1.2

ECDSA signatures and public keys are much smaller than RSA signatures and public keys.

Unsupported Ciphers

The following findings from Qualys SSL Labs provide an overview of known weaknesses in cipher suites used in SSL.

- Anonymous Diffie-Hellman (ADH) suites do not provide authentication.
- NULL cipher suites provide no encryption.
- Export key exchange suites use authentication that can easily be broken.
- Suites with weak ciphers (typically of 40 and 56 bits) use encryption that can easily be broken.
- RC4 is weaker than previously thought.[1] You should remove support for this cipher in the near future.
- 3DES provides only 108 bits of security (or 112, depending on the source), which is below the recommended minimum of 128 bits. You should remove support for this cipher in the near future.

References

1. IBM X-Force Threat Intelligence Quarterly 1Q 2014
2. Securing the mobile enterprise with IBM Security solutions – IBM 2013
3. Securing BYOD - CDW Reference Guide March 2013
4. Security Analysis of Amazon's Elastic Compute Cloud Service – Eurecom 2013
5. A Proper Foundation: Extended Validation SSL – Entrust 2013
6. Security Technologies for Mobile and BYOD – Kapersky 2013
7. Cloud Computing Vulnerability Incidents: A Statistical Overview – Cloud Security Alliance 2013

8. Overcoming the Security Challenges of the Cloud – CISCO/Intel 2013
9. Hypervisors, virtualization, and the cloud – IBM 2011
10. Security-As-A-Service – Alert Logic 2013
11. The Notorious Nine Cloud Computing Top Threats in 2013 – Cloud Security Alliance