Securing Defense Visual Information in a Commercial Environment

Defense Visual Information (DIMOC), Defense Media Activity (DMA)

Abstract

The Defense Visual Information (DIMOC) mission is to archive and distribute media assets to the Department of Defense (DoD), government, non-Federal agencies, commercial organizations, and general public. This necessitated an efficient low-cost solution to store and maintain records as well as to provide a delivery platform accessible to these aforementioned entities. DIMOC has entrusted a commercial system to accomplish this. Leveraging commercial capabilities has been a recent government paradigm in order to increase efficiency and reduce cost in an increased level of fiscal austerity. However it is equally important to consider how secure these systems are in handling potentially sensitive data. The consensus that the government needs to be more like business is equally countered by the need for business to be more like government in the area of cyber security. This paper discusses the challenges that DIMOC encountered in ensuring the security of DoD media in a commercially hosted environment.

Disclaimer

During the course of verifying contractor compliance with DoD security requirements, DIMOC found numerous vulnerabilities that did not conform to industry best practices. This statement is not intended to call out the contractor, but rather to illustrate the culture differences prevalent in the commercial sector despite recent high profile compromises like Sony, Ashley Madison, Target and others. The common denominator in all these attacks is that they could have been prevented easily by simply following basic security practices.

DIMOC's close partnership with the contractor in discovering vulnerabilities and cooperating with implementing solutions results in a positive outcome for both parties. DIMOC has assurance that its assets are protected and the contractor has a system it can count on to not only protect DIMOC assets but its other clients as well.

Background

DIMOC initiated a contract with a commercial media company for digitization of legacy analog physical holdings. The contract provided for a monetization effort by the contractor, and included having DIMOC provide current born-digital content (DoD-originated). Management of all the content is provided by the contractor's proprietary web applications. The contract included specifications for cyber security requirements based on DoD regulatory guidance [1].

The contractor promoted that it followed the Motion Picture Artists Association (MPAA) Site Security Program guidelines which offer best business practices to commercial media organizations. However, MPAA does not offer a certification although it is possible to have a third party provide an assessment using the guidelines.

An additional security factor is that DIMOC is considering revising the contract to include the ability to store Controlled Unclassified Information (CUI) assets. This would necessitate more stringent review of the contractor's security controls. Including CUI assets in the same system as the released assets simplifies asset management and discovery.

Presently the contractor systems are not actively processing CUI until security assurances are met; however, this does not negate the need for good security practices to ensure data integrity, availability, organizational reputation, non-repudiation and to protect against potential fiscal losses. An overview of the most significant challenges faced in confirming that the contractor achieves compliancy follows. Care is being taken to not release proprietary information and specific vulnerability data.

Requirements & Challenges: User Identification and Access

Identifying user and access group rules was one of the first requirements upon initiation of the contract. The commercial system was allowed to provide public access to released content while DIMOC maintained the requirement to allow access of released and unreleased content to DoD users only. DIMOC deferred management of public access accounts to the commercial contractor which simply implemented a standard username and password identification/authentication method. The contractor is obliged to ensure proper access controls are present on its system to ensure public users are not allowed to access un-released assets.

The contract required that DoD users must authenticate to the contractor's site via the DoD Common Access Card (CAC). The CAC is a highly secure authentication and encryption mechanism using the Public Key Infrastructure (PKI). Holders of the CAC perform two-factor authentication – they have something (the card) and they know something (the card's PIN). Because the contractor is required to allow all DoD members free access to our assets use of the card makes determining qualified users simple and error free.

The challenge became how could the commercial system and DoD CAC be integrated while incurring minimal cost to the government and the contractor? DIMOC already had CAC authentication capability using a Single Sign-On (SSO) solution. Given the state of the contractor's account management and authentication systems DIMOC determined that we would integrate the contractor's site into our SSO solution.

However, the contractor was not relieved of the requirement to harden their account management and authentication systems. This is because of the shared environment, DIMOC's assets reside in the same system as all other clients of the contractor. If another client's accounts could be easily compromised it puts DIMOC assets at risk. To meet this requirement the contractor purchased an identity management product and migrated all of their clients to that product.

DIMOC had to purchase seats for this product, then DIMOC's application developers worked closely with the contractor's developers and integrated DIMOC's SSO system with the contractor supplied identify management system, using the Security Assentation Markup Language (SAML)

The solution devised requires DoD users to visit DVI's web site where the user starts the CAC authentication process.

1. The user selects the "Search Released Imagery" hyperlink which sends them to the contractor's web site, using a link unique to DIMOC.
2. If no CAC is detected by the contractor's web site they are sent to the contractor's standard login page where further progress is not possible (if the user attempts a non-CAC login at that page the login will fail).
3. The user is then redirected to the DIMOC managed SSO server which authenticates the CAC and verifies the associated account.
4. If successful, the user is redirected back to the contractor's system with a SAML to allow access.
5. If unsuccessful, DIMOC's SSO login troubleshooting process begins.

It must be clarified that there are two account management systems linked via the SSO solution. DIMOC maintains a Lightweight Directory Access Protocol (LDAP) server that registers DoD CAC users to use its web applications and legacy asset management system. The contractor maintains its own account system to access it applications. During the SSO process the CAC identification number and email address are verified by the LDAP server against the registered account. Users may elect to register if they do not have an account with DIMOC.

Since the contractor's accounts are separately maintained new DIMOC accounts are automatically created in the contractor system using processes developed by the contractor, thus providing a seamless and effortless experience for the user while maintaining the highest degree of security (Figure 1).

Application Programming Interface (API) Authorization

APIs are used to facilitate the transport of metadata and bulk collections of files from DIMOC to the commercial systems as well as allowing asset re-use on other web sites. API calls between DIMOC and the contractor systems are normally made via the Hyper Text Transfer Protocol (HTTP), the same protocol that is used to view web pages and content on the internet. However, DIMOC requires that all traffic use the HyperText Transport Protocol Secure (HTTPS) both to prevent sniffing of traffic content and to provide a means of authenticating servers to one another ensuring the connection is expected and authorized.

An API call (where the external server makes a request of the contractor's server application) requires the external server to login. Such login, or authorization request for APIs, uses a hash token along with the account name (being machines they cannot exchange CAC information).

A hash is the mathematical product derived from known values from text or numbers. In practice this provides a unique set of data that is used by the contractor system to validate authorization to assets on its system. The hash used in the contractor's API code uses a combination of several known variables. The last component of the hash is the algorithm which acts as the "key" and is a value used to calculate the defined variables. The following is a basic conceptual example of a hash:

Variables x Hash algorithm (MD5, SHA2) = Hash output

A hash derived from a good hash algorithm is exceedingly difficult to break. Nonetheless two main concerns were discovered regarding the use of hashing. First, as stated previously, not encrypting the session traffic with HTTPS enables the traffic to be seen. While hashes may be hard to break, it is not impossible and having the hash output in the clear provides a potential attacker with information they should not have access to in the first place. An attacker can employ various means to attempt to crack the hash. Secondly, due to the way the hashing process was configured, large numbers of distinct API requests used the same hash output for authorization. This means that an attacker who can intercept API requests can change certain variables in the URL and resend it to the contractor system to gain access to a large number of distinct assets. This was due to the use of fixed hash variables that were used to produce the authorization string. The following example output will help illustrate this:

http://acme.com/vids/store/search/?keywords=Cats&view=deep&auth=dade%moneyTQ5dIbkv^3A905de0b0468e218

Breaking this down, "http://acme/com/vids/store/search/" invokes the search command. "?:keywords=Cats" is the search parameter. Directly following the search parameter is the authorization string, which is comprised of the username (dade), api key (money), followed by the hash output. In this example the hash variables are derived from the search command, username and api key which are all fixed and do not change, meaning the authorization string (which contains the hash) does not change for consecutive searches.

An attacker intercepting this can simply change the search parameter and reuse the authorization string and issue a command to gain access. Extending the variable parameter for the hash to include randomly generated data or simply using a more secure authentication and authorization method like public key can mitigate this issue. It did not help that these API's were sent in the clear in the first place so implementing HTTPS is an additional mitigation layer that must be applied.

API Vulnerability

In addition to the Authorization weakness for the API calls DIMOC discovered a vulnerability in the API itself. This vulnerability allowed users of the API (authorized and trusted users) to have access to other client's assets. This was caused by a failure to properly validate API requests.

After identifying this vulnerability to the contractor they performed their own assessment and agreed that DIMOC had found a previously unknown issue, and the contractor corrected the problem.

Web Site Security

HTTP versus HTTPS

DIMOC asset managers must use the contractor's web interface to curate assets and as such, upon examination of the main web interface, the following vulnerabilities were found:

The web interface was configured to use HTTP which as previously mentioned in the API discussion, is unencrypted and would allow eavesdropping of traffic between client and server. This issue was addressed by enabling HTTPS to applicable web

sites. This change affected all of the contractor's clients because of the shared environment.

Persistent Cookies

Persistent cookies are small amounts of text stored on the client computer, which are kept on the client computer permanently, or for very long periods of time. Cookies allow web sites to retain information about the user and their activity on their site to improve the user's experience. Cookies can be used to hold sensitive information, a practice that presents unacceptable risk to the user. DIMOC checked the contractor's use of cookies and found several problems.

It was discovered that persistent session authentication cookies were enabled and configured to a 12 month expiration date. This allowed users that logged into the site to have continued access for up to a year without requiring a fresh login. This is very convenient for the user, but without any parameters set for inactivity and enforced time-out this practice made the user's account vulnerable to session hijacking.

Session hijacking is a form of attack that exploits the active session for a third party to gain access to the site as if they are the currently logged in user. Setting an appropriate expiration date and time mitigates session hijacking by limiting the window of opportunity and requiring the user to re-enter their credentials. Sessions should expire every 30 minutes unless the user is active on the site.

It was also discovered that username and passwords were stored in persistent cookies. The username is stored in plaintext and the password in the form of a hash. This can give an attacker enough information to subsequently compromise the account. Conceivably this was done in order to make it easier for the system to remember credentials and allow quicker access to the web interface. According to the contractor's described best business practices, persistent cookies that store credentials in plaintext should not be used and access to content on websites should be set to automatically expire at predefined intervals [3].

Weak Password Management

The contractor system password policy was found to be inadequate. Password policies were not enforced in accordance to Motion Picture Association of America (MPAA) guidelines [4]. Specifically password complexity, account lockout thresholds and expiration were not configured. MPAA guidelines provide adequate guidance on implementing a solid password policy that includes strong password, password age, lockout threshold and history parameters that were not enforced.

Cross-site Scripting Vulnerabilities

Vulnerabilities in cross-site scripting were detected which can allow an attacker to inject malicious code, execute account hijacking and cause browser redirection. Unvalidated redirects and forwards were able to execute which made it possible for an attacker to redirect a user to a malicious site for phishing, to install malware, or bypass the application access control checks to gain administrative access. These vulnerabilities were found through the contractor via a third party assessor that conducts a review of their system on a yearly basis which was made available to the government. Like everything else described these were mitigated through proper configuration.

Incomplete Access Control Mechanisms

DIMOC discovered that assets from other clients could be accessed without any system authentication. Viewing DIMOC assets revealed the file-naming schema. With that information simply replacing known good values with random values revealed private images of other contractor clients. The weakness exposed was a result of decisions by the system designers to not require access controls for smaller image sizes. DIMOC pointed out that the image sizes in question were clearly adequate for web publication, and that the image content was clearly useful.

This particular issue is pervasive throughout the contractor's system and has yet to be adequately resolved, and is a significant barrier to authorizing the system to hold CUI assets.

Summary

The findings above came about despite the contractor's use of an external security review service. The requirements placed on the contractor pale in comparison to the full scale of information assurance requirements imposed on DoD systems.

The expression "Trust, but verify" is as concise a summary as could be given in the context of ensuring the protection of data with a commercial partner. It is ultimately the responsibility of the data owner (the client), to verify that proper security practices are implemented by the contractor. It is therefore important to establish these requirements from the beginning and include them specifically in the contract. This will define the clear expectations of the roles and responsibilities of both the client and the contractor.

However, computing technology is constantly advancing and new implementations and upgrades will inherently produce weaknesses and vulnerabilities indefinitely. Therefore it is important to periodically assess the security posture of hosting systems on a regular basis to refine and apply new security controls as required to better protect the client's assets. Planning for and sticking to a routine assessment program with the contractor should be a top priority for data owners.

References

[1] Department of Defense Instruction 8582.01, Security of Unclassified DoD Information on Non-DoD Information Systems, June 6, 2012.

[2] Single Sign-On Process Between Defense Visual Information (DIMOC) and Mass Digitization and Storage Contractor [sic] system, October 2013.

[3] MPAA Site Security Program, Client Portal, Control DS-13.6 and DS-13.67.

[4] MPAA Site Security Program, Authentication, Control DS-8.0.

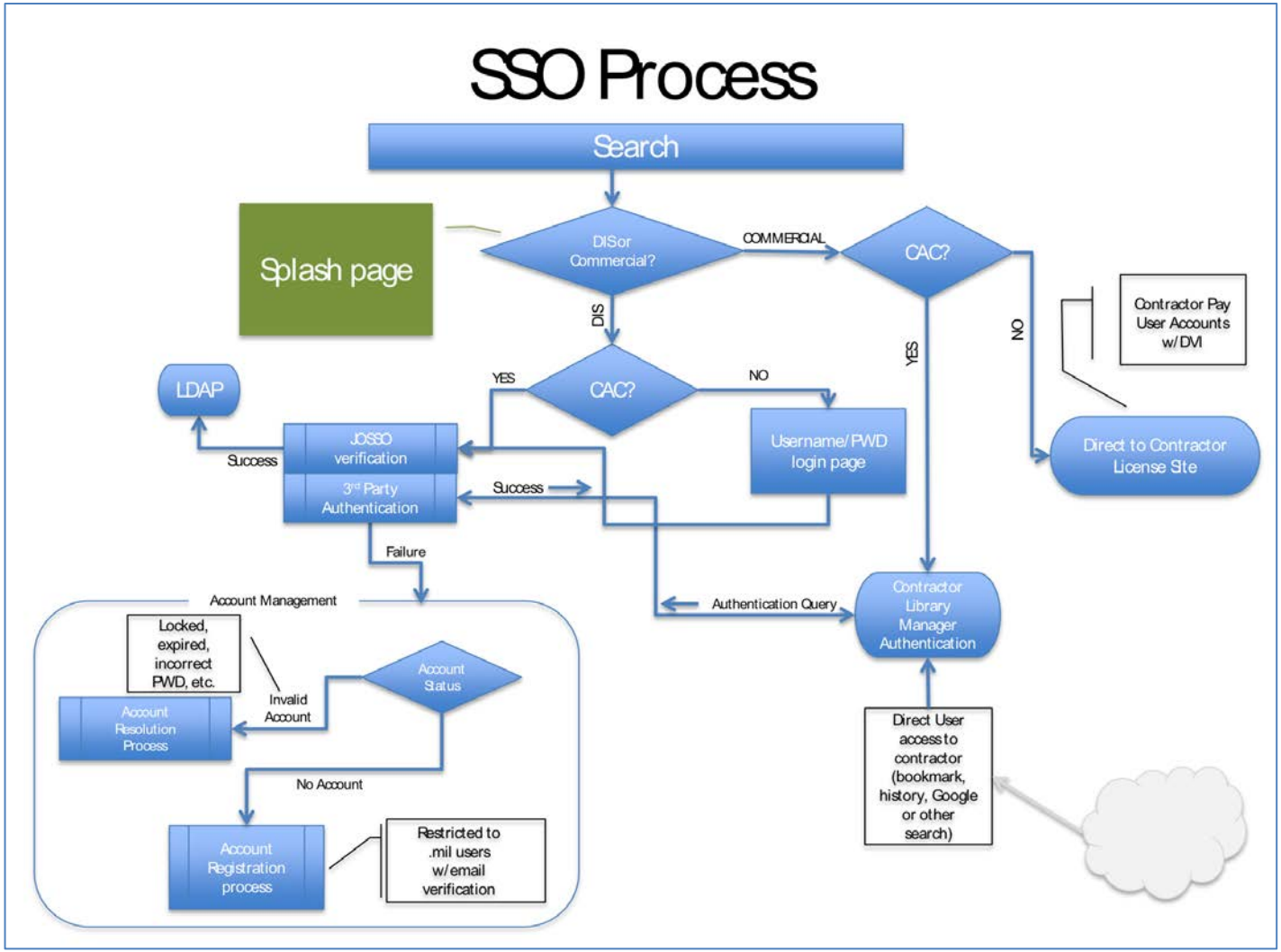

Figure 1: SSO process between DIMOC and contractor systems [2].