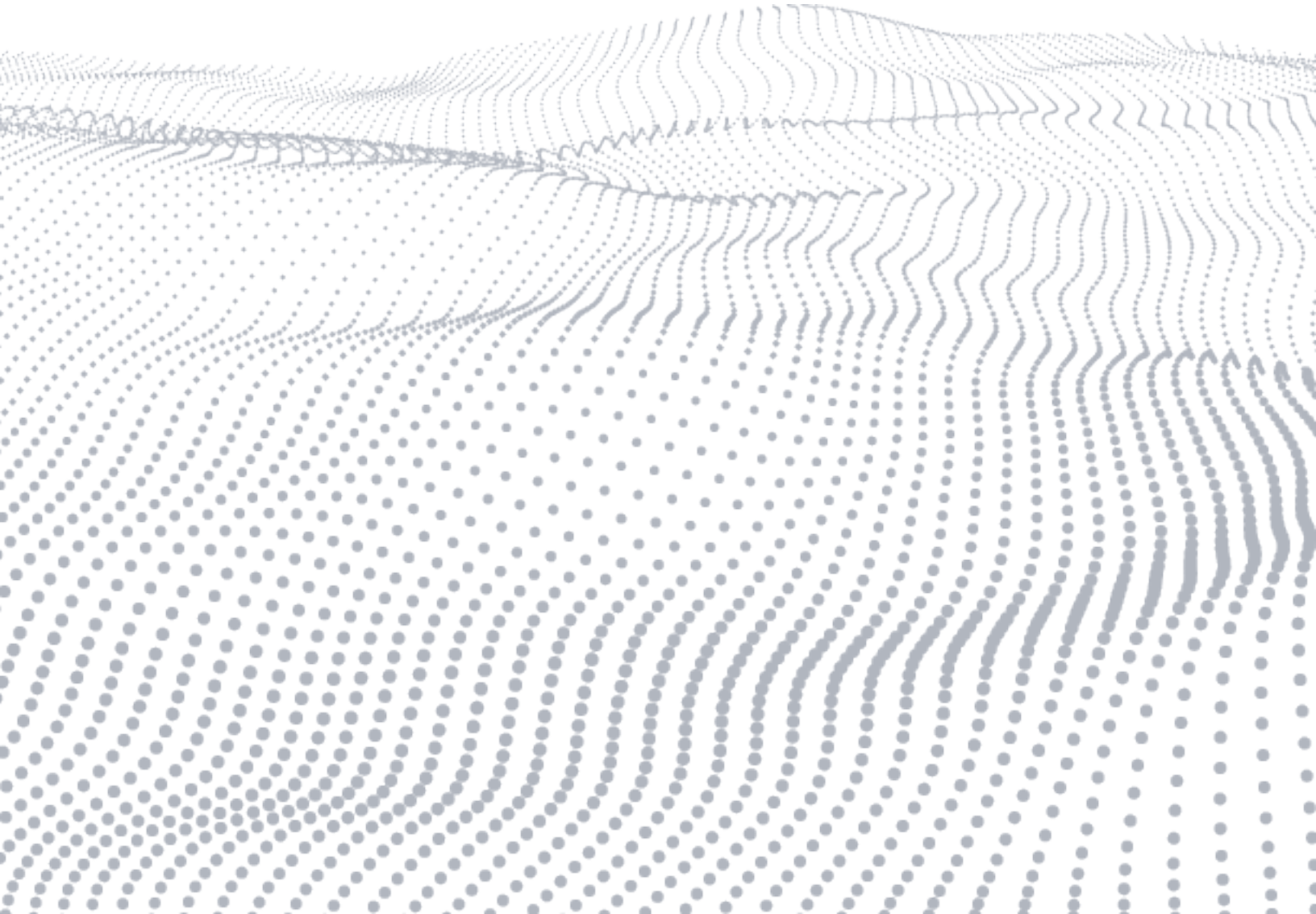
BLACKDUCK®

WHITE PAPER

Best Practices for Reducing Web Services and API Risks in M&A

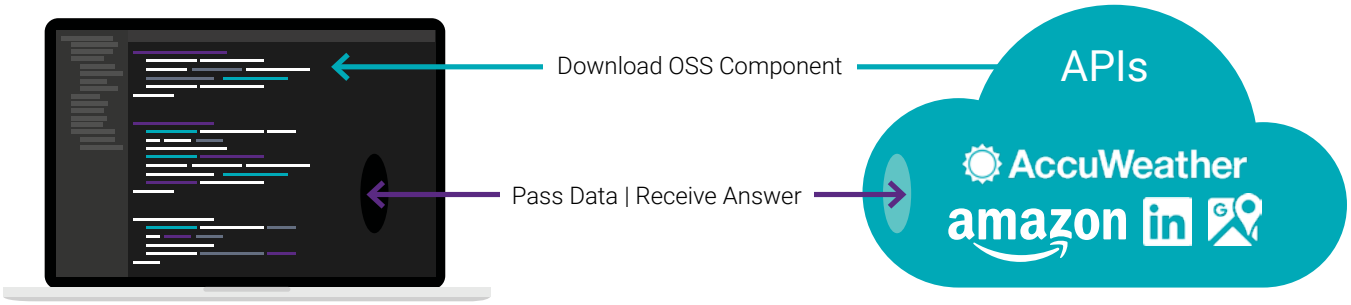


Web service and API landscape

A web service is an application or data source accessible via a standard web protocol, such as HTTP or HTTPS. Unlike web applications, web services are designed to communicate directly with other programs through application programming interfaces (APIs) and not directly with users via a browser.[1]

Today, applications may call out to any of tens of thousands of web services. Developers can easily employ these web services—usually for free—but companies often don't formally track the services that their applications call. For example, a developer could use the Google Maps API to determine latitude, longitude, and time zone data for display in an application without the development manager or the company's security team even knowing about it. It's worth noting that more and more companies are conscious of the need to manage the inbound APIs that they publish, but few manage or even track the "other side"—that is, the APIs their applications call.

As with open source software, web services and APIs raise issues of copyright, end user licensing, terms of use, and data and privacy policies. And if companies aren't tracking what their developers are using, they may be running afoul of legal requirements. For those looking to acquire companies that use web services and APIs, this adds a layer of risk to any M&A transaction. With the use of untracked, unmanaged web services and APIs on the rise, you must understand those risks to minimize them. And in tech M&A where software assets are a big part of the deal, you must minimize the risk by understanding what's in the software you're acquiring.



4 areas of web service and API risks

Within the area of web service and API risks, companies looking to conduct M&A activities have four primary sources of worry:

- Legal and compliance risk
- Security risk
- Data privacy risk
- Operational/business risk

Let's look at each type of risk a bit closer and examine their pitfalls, seemingly innocuous terminology, obfuscated cautions, misunderstood data, and other critical dependencies.

Legal and compliance risk

To confront the legal and compliance risks of web services and APIs, you need to identify their applicable agreements and terms. There are three issues to note:

First, you need to identify conditions of use to avoid copyright infringement. There really is no such thing as a "public domain API" that you can use freely without conditions to protect the IP of the API owner.

Second, those conditions are often complex and difficult to decipher. For example, the terms of use for Google APIs are described with over 7,000 words, not counting references and links to other documents.

Third, unlike terms of use for open source licenses, those for web services and APIs can be dynamic. The provider often reserves the right to change them at any time without notice and specifies that the user of the API agrees to those changes simply by continuing to use the service. An M&A target that uses APIs might have inadvertently agreed to these terms, via their developers, without full knowledge of their conditions. As the acquirer, you might inherit that responsibility.

To avoid legal and compliance risks with web services and APIs, companies must comply with the applicable agreements and terms. Compliance is important primarily to avoid breach of contract, such as by using the web service or API beyond the scope, or terms, of the agreement. Be careful, as there are often many details and nuances to the terms of agreements. Here are some common terms or actions the web service and API provider may undertake that you should watch out for:

- Patent nonassertion and defensive termination provisions
- No representations, warranties, guarantees, or SLAs regarding the API
- Changing the terms of or terminating your agreement at any time
- Suspending your access to the API at any time
- Deprecating or otherwise changing the API at any time

Security risk

When it comes to security risks of web services and APIs, there are two factors to consider. Not only do applications using APIs send data, information, or content to the web service provider, but they also receive data, information, or content. Most application owners focus on the data they are sending and not on what they are receiving. We'll come back to the receiving factor.

All the concerns about keeping customer and company data safe in the cloud or on-premises also apply to data sent to an API. Companies must be sure that there is a secure end-to-end transmission tunnel that cannot be eavesdropped on and that does not leak. Then they need to know how service providers treat the data once it reaches its destination. Do the service providers keep it encrypted? Do they send the data to yet another service provider?

Organizations have put less focus on received (or returned) data so far, possibly because there haven't been any highly publicized problems yet. But companies acquiring targets with applications that depend on web services and APIs will want to investigate the security of the returned data all the same. For example, an API could return executable code, binary files, images, or audio files (e.g., a text-to-speech API returns audio files). Or the API could return malicious code or embed viruses into returned files.

In addition, the data flowing in either direction could be intercepted. The application could be susceptible to man-in-the-middle attacks, where bad actors intercept communications between the company and the API service, either to eavesdrop or to secretly modify traffic between the two.[2] Many web services have the capability of encrypting data in transit—but some don't. In any event, ensure that encryption is turned on.

Real-world leaky API examples

- Salesforce had a REST API that leaked Marketing Cloud data for six weeks, but they weren't logging API calls, so they don't even know how many users were affected.[3]
- Google+, the failed social network, leaked the names, email addresses, ages, and occupation data of over 52 million users through its API for six days. In the end, fixing the leaky API was so difficult Google decided to discontinue the service instead.[4]

Data privacy risk

When considering the data privacy risk that web services and APIs pose to returned data, organizations should focus on incoming sensitive data, personally identifiable information (PII), and other data subject to data protection requirements. That's because some companies seem to treat data differently—or worse, fail to acknowledge its arrival—if it comes through the API side door versus the front door, where it would be subject to rigorous quality, safety, and security requirements before being accepted into a database. Because developers can use an API without the manager's or security team's knowledge, it's possible that the right people have not read the terms of the API agreement and do not know what data the API is returning.

Another risk to data privacy that API use can exacerbate is inadvertent data mashups, where the system combines API-returned data with other anonymized user data. When the API-returned data is linked to this anonymized data, someone may be able to deanonymize the aggregate data set and draw inferences about users that they are not allowed to.

Yet another risk to data privacy exists externally in the geographic locations where web services and APIs are physically situated. APIs are worldwide, so before a company uses one, they must understand where their data is going. Depending on the regulations of the jurisdictions where business is conducted, transmitting, receiving, storing, or processing data could violate the law (e.g., GDPR in the European Union).

To this point, our discussion of data privacy risks has concerned consuming API data. There is also a data privacy risk posed by publishing leaky APIs. An exposed public API that leaks may be inadvertently providing data to anyone interested in skimming it. To prevent this, API users must be certain that the data destination and method of transmission are secure, because senders are responsible for the data they send.

Real-world operational/ business risk examples

- When Spotify determined that SpotLister, a third-party playlist curator, had violated its terms of service, it deactivated SpotLister's API key, forcing the service to shut down.[5]
- When Twitter modified one API and deprecated another without releasing details about a replacement, Favstar, a third-party Twitter tracking service, had to shut down.[6]

Operational/business risk

When a company uses a web service or API in an application, they've created an operational dependency. Typically, there are no SLAs or guarantees when it comes to web services or APIs, constituting an operational and business risk. So an acquirer will want to ensure there is a contingency plan and/or set of redundant services in place in case the primary web service becomes unavailable for any reason.

And what happens if a large, well-resourced company acquires an organization that uses APIs supplied by direct competitors? Will the competitors shut down the APIs the acquisition depends on? What does this uncertainty do to the business? This potential disruption represents a business continuity problem and challenges the ability of smaller businesses to continue as a going concern. Keep in mind that the web service and API provider can suspend, terminate, or change the terms of access to the API at any time.

Understanding and mitigating web service and API risks

How should you deal with API risks when conducting software M&A activity? As with open source, the goal is to identify, quantify, mitigate, and allocate the risks of using someone else's software. In this case, it involves evaluating the use and consumption of third-party APIs. So what can you do to evaluate third-party APIs? There are three steps to this process.

Step 1: Identify web services and APIs

First, identify all inbound third-party—commercial or free—web services and APIs and related data or content accessed or used in the development, maintenance, support, and offering of products, as well as applicable licenses and usage statistics. What functions are the APIs calling? What do they deliver? If they deliver data, this delivery is subject to additional review, including determining whether the API service provider has the right to provide that data in the first place.

Second, and a more traditional concern, it's worth understanding the public APIs published by the target and how they are managed and secured. This consideration is beyond the scope of this paper.

How can you identify all the inbound and outbound web services and APIs that your target uses? You can start with self-disclosure. Ask their developers to compile a list of APIs in use by conducting a manual string or keyword search for API call syntax in the codebase. For any APIs they pay to access, conduct an audit of their procurement records, if any. This process is analogous to obtaining a bill of materials for open source and third-party software components.

But the problem is that most companies don't track what API calls their developers have incorporated into their applications, so you are unlikely to get correct and comprehensive answers. To address this shortfall, you can conduct code scans with a web services audit (e.g., from Black Duck Audit Services) that will generate a list of the API calls in the codebase. This information will help you compile the bill of materials, what data the APIs deliver, and how that data is used, and will help you infer how important the APIs are in the codebase.

Step 2: Analyze to understand what you have

Subject to legal protection, you must analyze the contract terms of the web services and APIs used to ensure they cover the target's use. For example, using the free Google Maps API to display maps in a commercial product would be incompatible with Google's terms of service, or contract, which stipulates the free Maps API can only be part of a free, publicly available service.[7] You must do this analysis on two levels to understand the applicable terms and any:

1. Incompatibilities between the described or proposed use of a given web service/API and the target's actual use
2. Differences between the data or content described in the contract terms and the data or content actually provided by the web service/API (which is separate and protectable)

Also analyze the business terms of the web service, which may be incompatible with your current or proposed business practices. This analysis will help you understand whether any data or content provided by the web service/API is out of compliance with internal policies or applicable law.

Finally, grasp the importance of the functionality of the data or content provided by the web service/API. For example, if your target is a social media analytics company and three of the 20 APIs they use are critical (e.g., Facebook, LinkedIn, Twitter), what happens if those three APIs go away? How do you mitigate that risk? Can the target enter into special agreements with the API providers? Can the target clarify that they're operating within the current terms of those APIs? This analysis ties into how the business may be impacted by the terms of use for web services and APIs.

Step 3: Create a remediation plan

To mitigate risks related to web services and APIs, you must create a remediation plan to address identified issues that fall into one or more buckets:

- Code remediation
- Legal remediation
- Redundancy/contingency plan remediation
- Data privacy remediation
- Risk allocation (through contractual terms)

Code remediation. Code remediation involves removing or replacing web services and APIs if they are interrupted or shut off. This is the bucket where you will address identified security vulnerabilities, defining the cost to remediate in terms of engineering resources and time on task.

Legal remediation. Under legal remediation, you may need to seek or amend a software license for an API because there may not be a specific API addendum for the software you need. For example, you may need to negotiate an API SLA or otherwise ensure that the API will be available for a definite period of time. And you'll also want to have the grounds and procedures for terminating an API license spelled out. Other clarifications could also be necessary.

If you find out in your due diligence that the target used the API improperly, without permission, or without paying for it (if applicable), you must seek waivers of past liability for copyright infringement. Of course, when the current agreement expires, you will need to relicense the API or obtain a new API license. Customary outside legal expenses, inside counsel time, and fees to licensors all apply.

Redundancy/contingency plan remediation. This type of remediation is especially relevant once the acquisition has closed. You'll need alternative API solutions ready to go into place when—not if—mission-critical APIs fail. Or you may determine that you need additional redundancy once you've begun to operate the acquisition yourself.

Data privacy remediation. Monitor and audit the M&A target's use of web services and APIs to identify the nature and scope of the data being collected and transferred, as well as the data being received. This audit includes identifying where that data is collected and where it is going. With this information, you can properly assess the risks associated with such APIs and, ultimately, tell your user base what information is actually being collected and transferred, as well as confirm that the data-transfer channels are secure.

Risk allocation (through contract terms). Particular to transactions on the buy side, you need representations and warranties to allocate the risk that what the seller is selling is in fact what they say it is, and if it isn't, to provide a remedy for the misrepresentation. For example, the seller should represent that they are complying with the terms of service governing the APIs in question. You can also have remediation-focused closing conditions and best-efforts covenants to get certain things cleaned up by closing. Specific indemnities and additional escrows can serve as insurance that these issues are remediated in time.

However, note that all these contractual terms for allocating risk will run out. The seller eventually wants to get their money out of the transaction, and the contractual terms serve as the guide to the end game regarding how things will be fixed. At the conclusion of the transaction, you will be the owner of the company and its codebase, with all the express and implied realities that go along with that.

Impact on M&A and other transactions

Due diligence requests have become commonplace for open source software, and the same thing is happening more often with APIs. We now see due diligence requests covering APIs and web services and data in acquisitions. People are becoming aware of it and asking for it.

Due diligence for APIs can be swept into scheduling and substantive reps in different ways, often by simply adding the words "APIs," "web services," and "accessed by" to the definition of "third-party or open source software." Or you can add the definition of "third-party platform," which would include any application, website, or software-as-a-service (SaaS). Then you can sweep this into your due diligence request, saying that a third-party platform is anything that the target is accessing, linking to, or getting data or content from. This description will cover any API or web service without using that terminology. Other impacts APIs can have on M&A and other transactions:

- Need for additional reps specifically addressing APIs and interfaces, use of or access to web services or SaaS, and data and content interoperability and compatibility
- Remediation covenants and closing conditions
- Indemnities and escrows

Executive summary

- The use of web services and APIs is rapidly rising.
- Most companies don't track or manage APIs (same as open source).
- APIs pose risks similar to those from open source.
- The Black Duck Web Services and API Risk Audit helps identify web services and APIs in use and outlines the risks associated with them.

Learn more about the Black Duck® Web Services and API Risk Audit

Due diligence concerning web services and APIs is complex. You can try to conduct your due diligence manually in a time-consuming, hit-and-miss manner, or you can use an automated service powered by machine learning to help you.

The Black Duck Web Services and API Risk Audit gives you a listing of the external web services used by an application, with insight into potential legal and data privacy risks. The summary report allows you to quickly evaluate web service risks across three key categories: governance, data privacy, and quality. To learn more, read the Black Duck Web Services and API Risk Audit datasheet. No registration required.

Endnotes

1. TechTerms, Web Service Definition, updated Aug. 10, 2017.
2. Dan Swinhoe, What Is a Man-in-the-Middle Attack? How MitM Attacks Work and How to Prevent Them, CSO, Feb. 13, 2019.
3. Tara Seals, Salesforce.com Warns Marketing Customers of Data Leakage SNAFU, Threatpost, Aug. 3, 2018.
4. Janet Burns, Google Plus to Shut Down Early After API Bug Fumbles 52 Million Users' Privacy, Forbes, Dec. 10, 2018.
5. Austin Powell, Spotify Shuts Down Major Third-Party Service After Playlist Scandal, Daily Dot, March 17, 2018.
6. Patricio Robles, API History Repeats Itself as Recalibrated Twitter Dev Strategy Forces Shutdown of Favstar, ProgrammableWeb, May 17, 2018.
7. Google, Google Maps Platform Terms of Service, Section 3: License, updated Nov. 21, 2019.

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.