Apples to Oranges: Understanding the Changing Attack Surface for Applications Migrated from Self-Hosted to SaaS

Author: Eddie Black, edd.n.black AT gmail.com
Advisor: *Clay Risenhoover*

Accepted: *13 August 2023*

Abstract

"Comparing an on-premises application to a SaaS application is like comparing apples to oranges." – Anonymous SANS MSISE faculty.

Why would you defend two instances of the same application differently? Self-hosted applications have the same security concerns as SaaS applications, but they cannot be protected the same way. To move a hosted application to a SaaS provider gives the illusion of offloading risk. Many organizations do not know what visibility they are giving up, what controls they are still responsible to implement, and how to continually verify the provider is properly mitigating the offloaded risk. In this paper, the author will examine the self-hosted and SaaS versions of the Atlassian applications Confluence and Jira. The author will compare the logging and event structure between self-hosted and SaaS versions of both applications to identify any gaps that exist, verify those gaps line up with the expected gaps in the MITRE ATT&CK Enterprise and Cloud matrices, and identify whether current research adequately covers those gaps. The author will then discuss methods for identifying and mitigating the remaining gaps.

1. Introduction

Modern organizations run on applications. As rapidly as application delivery has changed from single server hosted, to virtual server hosted, to cloud hosted, discussions on delivery methods have gone back over a decade. Cloud computing – "common location independent, on-line utility on demand" (Cervone, 2010, p164) has now existed long enough that the cloud hosted server has started to give way to serverless code applications through event-driven code hosting such as AWS Lambda (Srinimalan, et al, 2019, p412).

Securing applications has always been secondary to mining value from the data manipulated by the application (Culot, et al, 2020, p76). Security can best be viewed as risk mitigation – its cost is justified by the cost likelihood of the security incidents it prevents. Applications have become the "top action vectors in breaches" and the "Top action vectors in incidents" (Hylender, et al, 2023, p4). Widely adopted standards that have been developed for Information Security programs are designed to be general to encompass as many organizations and scenarios as possible (ISO 27001, 2022, p1). ISO 27001 describes various security controls that must be implemented to be considered compliant with the standard. Some of these controls designed for a traditional self-hosted environment (e.g., Threat Intelligence, Access Control, Information security during disruption) take on new bounds in a cloud environment. The ISO 27001 standard has separate requirements to address security explicitly in cloud environments (e.g., Information security in supplier relationships, Information security for use of cloud services) that extend compliance requirements (ISO 27001, 2022, p11-13).

This author delves into the differences in application security between applications hosted in a customer-controlled environment (whether physical or virtual) or and a vendor hosted (software as a service) environment. These differences tie to the new control layering introduced by ISO. The author will be using Atlassian products with both a self-hosted and Atlassian hosted version - in this case Atlassian Confluence and

Atlassian Jira. The focus is to identify gaps in visibility between the same applications, differences in needed mitigations, which mitigations are cost effective for that amount of risk, and the risks that cannot be mitigated. In the view of the author, it is always better to know a risk exists that one cannot mitigate, than to be ignorant of its existence.

2. Research Method

2.1. Scope

Evaluating security differences between self-hosted and SaaS applications can have an expansive scope. The author had a constrained research timeframe and found it best to limit the scope to consumed applications rather than published applications. A consumed application is one where an organizations users make use of the application for their own needs. A published application is one that an organization provides to outside users and organizations so those external users can make use of the application for their own needs.

The scope of this project focuses on two applications with both self-hosted and SaaS versions. The Atlassian applications Confluence (a wiki application) and Jira (a ticketing application) were selected for this. The author saw that two applications had similar audit designs for event generation. In addition, the author has access to audit event logging for both applications in both environment types.

The scope also limited how to compare the security measurements. The MITRE ATT&CK framework has 17 different matrices for mapping adversary attack techniques (MITRE, 2023). The author chose to use the MITRE Enterprise Matrix and the MITRE Cloud SaaS Matrix for comparing attack vectors against the applications, and in the case of self-hosted, the infrastructure on which it is hosted.

The final part of the scope is addressing solutions to a common governance framework. The author selected the well-known ISO 27001 standard updated in October 2022 which includes compliance sections for SaaS applications.

2.2. Application Environment

The two on-premises versions of the applications were hosted on Windows servers. Then the two SaaS versions of the applications were hosted by the vendor in a public cloud. Based on DNS resolvers for the SaaS application's login pages, the Atlassian SaaS applications were hosted in an Amazon Web Services (AWS) environment. The author could not determine if there was any part of the SaaS application or environment that was hosted in a different cloud, e.g., Google Cloud or Microsoft Azure. Based on this, the author operated under the assumption that the application was hosted exclusively in AWS for the sake of this research.

2.3. Research Approach

The author pulled 30 days of application logs from each of the four applications. In addition, the author made use of Atlassian's published Audit Event Catalogs (Atlassian, 2023) to see further events that exist in the catalog that were not generated over the course of the 30-day period. These events were mapped against the various techniques in their relevant MITRE ATT&CK Matrices. These techniques were then compared to the ISO 27001 standard to check how the application detections mapped to the control standard.

3. Findings and Discussion

An information security comparison of self-hosted applications versus their SaaS counterparts requires a series of logical steps. First, the author will look at the differences in the MITRE ATT&CK Cloud SaaS and MITRE ATT&CK Enterprise frameworks to identify the differences in the attack landscape. Second, a comparison is made of 30-day batches of logs of both versions of Atlassian Jira and Atlassian Confluence applications. Third, analysis of the event mapping to the MITRE Frameworks highlights the gaps in coverage and highlights the areas of concern. Fourth, analysis of the blind spot created by the loss of visibility in the infrastructure illustrates the risk needing to be mitigated. Fifth, the findings are summarized to show the action items that organizations should be aware of and address per commonly used standards such as ISO 27001.

3.1. MITRE Differences

The MITRE ATT&CK framework is a collection of attack techniques that line up with the steps an adversary will take over the course of an attack (MITRE, 2023). The main framework, the MITRE Enterprise framework, consists of 14 tactics and 233 techniques, many of which are divided into multiple sub-techniques.

Tactic	Number of Enterprise Matrix Techniques	Number of Cloud Matrix Techniques	Number of Cloud SaaS Matrix Techniques
Reconnaissance	10	0	0
Resource Development	8	0	0
Initial Access	9	5	4
Execution	14	4	1
Persistence	19	7	4

Privilege Escalation	13	3	2
Defense Evasion	42	9	3
Credential Access	17	9	7
Discovery	31	13	4
Lateral Movement	9	3	3
Collection	17	5	3
Command and Control	16	0	0
Exfiltration	9	2	1
Impact	13	8	3
Total	**233**	**68**	**35**

Figure 1 – Count of MITRE ATT&CK Techniques by Tactic and Matrix

The data that stands out is the dramatic decrease in the count of available attack techniques mapped to adversary behavior for a SaaS environment. A team might initially conclude that SaaS applications have a dramatically smaller attack space, and that application security is by default better by using a SaaS solution.

To understand the difference between environmental risks of traditional versus SaaS applications, the audience needs to understand the cloud shared security model. The cloud shared security model designates which groups is responsible for the security of what part of the infrastructure (Tank, et al, 2019). In basic terms, the service provider (AWS, Azure, or Google) is responsible for the security of the hardware and the provided infrastructure (back-end networking, servers, compute). The customer is responsible for securing the data and methods of access to that data, the transport of the data, and anything else that is hosted within the cloud provider's Infrastructure as a Service (IaaS)

offering. There is a common misunderstanding that offloading the risk mitigates it. It does not.

The SaaS model should not be confused with the PaaS model, or the IaaS model. In the case of the Atlassian apps, there are multiple layers to consider. Per Figure 1, the count of attack techniques shrinks dramatically when going from Enterprise to Cloud, and then further shrinks when looking at SaaS. This ties back to the shared responsibility model, where Atlassian is in the role of the customer, and AWS is in the role of the provider. Amazon provides the IaaS layer, and likely the PaaS layer, then Atlassian lays the SaaS layer on top of the AWS controlled layers. The shared responsibility model divides up who is responsible for security of a given component. In many cases, the use of services at a given layer from a vendor comes with an agreement that has some level of indemnification in case of an issue with the vendor. This creates a forced trust. A customer must trust that the vendor is taking proper steps to secure the layers of the service model that include the components being offloaded. An organization gives up visibility and control of lower-level service components while believing they are offloading risk. The consequences of that trust being violated may only be negligible damage to the vendor's brand. The risk may only look offloaded, but a given team in an organization still bears the risk if they choose a SaaS platform that causes operational impact, no matter where in the shared responsibility model the issue occurs.

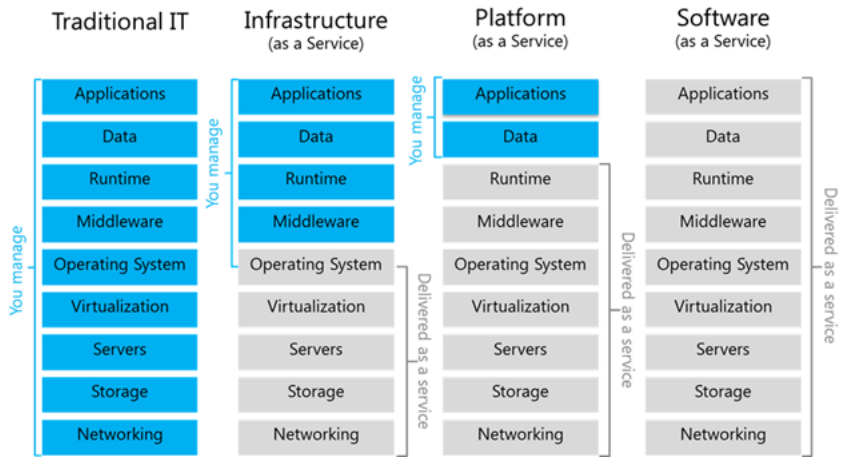

Figure 2 – Cloud Service Models (Chou, 2018)

3.2. Atlassian Jira Event Comparison

Atlassian Jira breaks down its audit logs into six main categories (Atlassian, 2023):

- Global configuration and administration
- User management
- Permissions
- Local configuration and administration
- Security
- End user activity

These categories cover the entirety of the cloud event catalog for the Jira application. Depending on the age of various SaaS applications, their self-hosted versions may not have published event catalogs. Having the same event types in the same categories in each version does not necessarily lead to identical logs. Taking one of the most common events seen in the period of review, User Added to Group, the logs are similar in format but not identical.

Self-hosted Jira (the data in brackets has been sanitized for privacy):

{LogStreamName: <AWS CLOUDTRAIL IDENTIFIER>,Timestamp: 1689353962673,Message: {"affectedObjects":[{"name":"<GROUP NAME>","type":"GROUP"},{"id":"<Jira User System Identifier>","name":"<NAME OF TARGET USER>","type":"USER"}],"auditType":{"action":"User added to group","actionI18nKey":"jira.auditing.user.added.to.group","area":"USER_MANAGEMENT","category":"group management","categoryI18nKey":"jira.auditing.category.groupmanagement","level":"BASE"},"author":{"id":"-1","name":"System","type":"system"},"changedValues":[],"extraAttributes":[],"method":"Unknown","system":"<APPLICATION URL>","timestamp":{"epochSecond":1689353960,"nano":16000000},"version":"1.0"},IngestionTime: 1689353967695,EventId: <APPLICATION EVENT ID>}

Cloud Jira:

{"id":<APPLICATION EVENT ID>,summary":"User added to group","created":"2023-07-14T16:33:41.950+0000","category":"group management","eventSource":"","objectItem":{"name":"<TARGET GROUP>","typeName":"GROUP","parentId":"10000","parentName":"com.atlassian.crowd.directory.IdentityPlatformRemoteDirectory"},"associatedItems":[{"id":"<USER ID NUMBER>","name":"<USER ID NUMBER>","typeName":"USER","parentId":"10000","parentName":"com.atlassian.crowd.directory.IdentityPlatformRemoteDirectory"}]}

The first noted difference in the self-hosted (via IaaS) is that the logs are funneled through AWS Cloudtrail and receive Cloudtrail log markers. Other differences in the SaaS version include having human readable timestamps, and that the SaaS version shifts from usernames to system assigned alphanumeric identifiers for the logs.

Over the 30-day period of this research, Jira Cloud produced more unique events than self-hosted Jira. Most of this is due to the organizations greater use of Jira Cloud, generating category update events, rather than the category added or deleted events found in both systems. The captured event names and counts for the Jira events can be found in Appendix A.

3.3. Atlassian Confluence Event Comparison

Atlassian Confluence breaks down its audit logs into six main categories (Atlassian, 2023):

- Global configuration and administration
- User management
- Permissions
- Local configuration and administration
- Security
- End user activity

These categories cover the entirety of the cloud event catalog for the Confluence application. Like Jira, Confluence has overlap in the data similar logged events. Once again, the event User Added To Group is being used as an example.

Self-Hosted Confluence (the data in brackets has been sanitized for privacy):

{"affectedObjects":[{"id":"<"OBJECT ID>,"name":"<OBJECT NAME>,"type":"Group"},{"name":"<GROUP NAME>,"type":"User"}],"auditType":{"action":"User added to group","actionI18nKey":"audit.logging.summary.group.membership.added","area":"USER_MANAGEMENT","category":"Users and groups","categoryI18nKey":"audit.logging.category.user.management","level":"BASE"},"author":{"id":"-1","name":"System","type":"system"},"changedValues":[],"extraAttributes":[],"method":

"System","system":"<APPLICATION "URL>,"timestamp":{"epochSecond":1677347059,"nano":816000000},"version":"1.0"}

Confluence Cloud (the data in brackets has been sanitized for privacy):

{"author":{"type":"user","displayName":"System","operations":null,"isExternal Collaborator":false,"accountType":"","publicName":"Unknown user","externalCollaborator":false},"remoteAddress":"","creationDate":1689356175849 ,"summary":"User added to group","description":"","category":"Users and groups","sysAdmin":false,"superAdmin":false,"affectedObject":{"name":"<GROUP "NAME>,"objectType":"Group"},"changedValues":[],"associatedObjects":[{"name":" <"USER>,"objectType":"User"}]}

This time, the self-hosted application did not pipe its logs through AWS Cloudtrail, so the associated metadata is not included in the logs. With Confluence, the objects have a name, but only in the self-hosted version is there an ID field separate from the object name field. Other than these minor differences, the log content is nearly identical. The captured event names and counts for the Confluence events can be found in Appendix B.

3.4. SaaS Customer Blind Spots

The logs generated by the application can map to some of the 35 MITRE Cloud SaaS Matrix techniques. That leaves a lack of visibility into 198 techniques from the MITRE Enterprise Matrix. The difference is self-hosted applications have a wealth of additional logging from different systems to address these additional tactics and techniques. Netflow logs can help address reconnaissance against the infrastructure. System logs can help detect resource development within the environment (e.g., deploying new instances or enabling services). Systems and netflow can capture the creation of command and control channels via scheduled communications and processes.

MITRE removed the tactics of reconnaissance, resource development, and command and control from the Cloud SaaS Matrix. Some reconnaissance can still be

done. An adversary can still use reconnaissance techniques like: Gather Victim Identity Information, Gather Victim Org Information, Phishing for Information, and Search Open Websites / Domains. An adversary can do the same organization and user reconnaissance that is done for enterprise attacks to build out some of the information to brute force and organization's SaaS portal. Phishing can allow an adversary to trick an organizations SaaS administrator to grant access to external users or applications, actions which will be logged by the supplication. Adversaries can discover which SaaS platforms an organization uses by building potential subdomain lists and using them to query that subdomain with popular SaaS applications to find login portals and application programming interface (API) endpoints.

Accounting for the techniques that can still be seen and managed that do not appear on the Cloud SaaS Matrix, that still leaves over 185 techniques to which any SaaS using organization in blind. Common practice is to request a SOC II or ISO 27001 certification. These documents are usually provided upon request should an organization make a governance request during the initial analysis phase. However, these attestations go one level deep. A SaaS customer will see attestations for the SaaS vendor, but not the underlying PaaS or IaaS vendors. Once again, a degree of trust is placed in the SaaS vendor that they are doing their due diligence down the stack.

High profile supply chain breaches are occurring more often. Breaches have occurred through large vendors including Solarwinds (Jibilian, 2021), Kaseya (US DNI, 2021), Microsoft (Raina, 2021), and Atlassian (Abrams, 2023). In three cases, supply chain breaches were identified by the direct impact against the customers. When these breaches happen, customers are at the mercy of the vendors for timely and accurate reporting of the exposure. This lack of visibility is frustrating for organization that do everything they can to secure their operations, only to have a trusted vendor come up short. Incidents like these only amplify blind spots and bring into focus organizations

priorities between the various facets of security: confidentiality, integrity, and availability.

3.5. MITRE Cloud Mapping

There are known blind spots based on the MITRE Matrices. However, just because a technique exists within the Cloud SaaS Matrix, it does not automatically mean that the SaaS application is vulnerable to the technique. By mapping the techniques to application capabilities and logging, gaps can be identified on the SaaS consumer's side of the shared responsibility model. These gaps are the easiest to identify and mitigate. All the following techniques will reference the MITRE ATT&CK Cloud SaaS Matrix (MITRE, 2023) and the events referenced come from the Confluence event catalog and the Jira event catalog (Atlassian, 2023). Some of the MITRE ATT&CK Techniques are relevant to multiple Tactics. Any repeat appearance of a technique will only be addressed only if the tactic will generate different events.

3.5.1. MITRE Tactic – Initial Access

The initial access tactic is broken down into four main techniques.

Technique Identifier	Technique Name
T1189	Drive-By Compromise
T1566	Phishing
T1199	Trusted Relationships
T1078	Valid Accounts

Figure 2 – Initial Access Techniques

Drive-By Compromises either work by pushing an exploit into the SaaS application through the browser or capturing in browser credentials or tokens. Software exploits attack the application itself, and any events triggered will be determined by the nature of the exploit.

Phishing against a SaaS application attacks the users and is not a direct attack against the application. Trusted relationships revolve around add-ins and generate the same events: Confluence Authentication and Jira Login. These events have source IP addresses in the events, which can be tracked and alerted on based on the username and source IP address combination.

3.5.2. MITRE Tactic – Execution

The execution tactic has one technique.

Technique Identifier	Technique Name
T1648	Serverless Execution

Figure 3 – Execution Techniques

This technique uses existing services to execute code in the application. Confluence does not have a setup for this, as the only automated services are all for administering the workspace and not for deploying any code. Jira has workflows which can be set up to run specific processes. A crafty adversary can hide code in the workflows set to specific triggers. Jira logs workflow events. Based on the design of the Jira system, this technique is one that will be hunted, rather than have specific alerts built for it.

3.5.3. MITRE Tactic – Persistence

The persistence tactic is broken down into four techniques.

Technique Identifier	Technique Name

T1098	Account Manipulation
T1546	Event Triggered Execution
T1556	Modify Authentication Process
T1078	Valid Accounts

Figure 4 – Persistence Techniques

Persistence mechanisms are abused in several ways. Modification to an authentication process triggers several authentication alerts in both applications. The events involve changes to password policy and the use of Multi-Factor Authentication (MFA). Any unscheduled changes to those should generate a critical alert. A Jira workflow event would flag on the event triggered execution. There is no parallel on the Confluence side. Account manipulation should only source from documented administrator accounts. The creation of new accounts should tie to processes around the addition of new users. All this behavior ties to user creation and modification alerts in both applications.

3.5.4. MITRE Tactic – Privilege Escalation

The privilege escalation tactic is broken down into two techniques.

Technique Identifier	Technique Name
T1546	Event Triggered Execution
T1078	Valid Accounts

Figure 5 – Privilege Escalation Techniques

These techniques trigger the same events that show up in persistence. With this tactic, security permissions events will trigger in both applications. Permissions to different workspaces in both applications should be reviewed periodically. Any

escalation to privileged levels of access should generate an alert to be checked for legitimacy.

3.5.5. MITRE Tactic – Defense Evasion

The defense evasion tactic is broken down into three techniques.

Technique Identifier	Technique Name
T1550	Use Alternate Authentication Method
T1078	Valid Accounts
T1556	Modify Authentication Process

Figure 6 – Defense Evasion Techniques

The use of alternate authentication methods is a common attack vector. Most administrators are aware of access through the web portal. API endpoints may not require the same level of security. Both applications log API authentication the same way standard user authentication is logged. Event alerting can be built to look for API authentication events sourcing from unexpected IP addresses.

3.5.6. MITRE Tactic – Credential Access

The credential access technique is broken down into seven techniques.

Technique Identifier	Technique Name
T1110	Brute Force
T1606	Forge Web Credentials
T1556	Modify Authentication Process
T1621	Multi-Factor Authentication Request Generation

T1528	Steal Application Access Token
T1539	Steal Web Session Cookie
T1552	Unsecured Credentials

Figure 7 – Credential Access Techniques

Authentication events in both systems capture most of this behavior. Brute force is a common alert looking for a threshold of failed login or authentication events. In these systems, web credentials can be forged, but that would require administrative access which should already be monitored, as well as the creation of special credentials like API tokens. These applications do not have logging for MFA requests. MFA is handled through Single Sign On (SSO) platforms that have their own logging. Cookie access is not being monitored as authentication is being managed by session tokens. Those tokens generate authentication and system events who the token transfers users or systems. Unsecured credentials can be tracked through alerts built around user behavioral analytics. With ticketing and wiki applications, periodic audits should be done to make sure credentials and tokens are not stored in pages or workflows.

3.5.7. MITRE Tactic – Discovery

The discovery tactic is broken down into four techniques.

Technique Identifier	Technique Name
T1087	Account Discovery
T1526	Cloud Service Discovery
T1069	Permission Groups Discovery
T1518	Software Discovery

Figure 8 – Discovery Techniques

There are two ways these events show up in the applications. First is through administrative access. These events are generated following the administrator account's behavior accessing different workspaces. They look benign as part of the administrator's work. For a non-administrative user, Jira tracks discovery through object event searches and Confluence tracks the events as end user searches. Non-administrators generate permission denied events when trying to view permissions.

3.5.8. MITRE Tactic – Lateral Movement

The lateral movement tactic is broken down into three techniques.

Technique Identifier	Technique Name
T1534	Internal Spearphishing
T1080	Taint Shared Content
T1550	Use Alternate Authentication Material

Figure 9 – Lateral Movement Techniques

As consumers of SaaS content, lateral movement does not apply unless the consumer is a reseller of the SaaS applications putting a management layer over it, acting as an administrator of the SaaS application for their clients. Authentication alerts would be seen across instances of both Jira and Confluence. Those events would be best seen if the events were forwarded to a log collector so access across instances can be correlated.

3.5.9. MITRE Tactic – Collection

The collection tactic is broken down into three techniques.

Technique Identifier	Technique Name
T1119	Automated Collection
T1530	Data from Cloud Storage

T1213	Data from Information Repositories

Figure 10 – Collection Techniques

There is no automated collection for these applications. Nor does a cloud storage attack apply. Both applications are information repositories, and the collection would only generate events for workspace or project access or changes. Alerts can be generated based on a quantity threshold of events tied to a single user's workspace and project access indicating the staging of data for exfiltration.

3.5.10. MITRE Tactic – Exfiltration

The exfiltration tactic has one technique.

Technique Identifier	Technique Name
T1048	Exfiltration Over Alternate Protocol

Figure 11 – Exfiltration Techniques

This does not apply. Both applications were designed to not need access over different protocols (e.g., SSH, FTP, Telnet). The access is all handled over HTTPS, although Confluence allows the use of WebDAV for the transfer of large files. Any imports or exports generate events: data pipeline events in Jira and data pipeline and import / export events in Confluence.

3.5.11. MITRE Tactic – Impact

The impact tactic is broken down into three techniques.

Technique Identifier	Technique Name
T1531	Account Access Removal
T1499	Endpoint Denial of Service
T1498	Network Denial of Service

Figure 12 – Impact Techniques

Any changes to account access, whether it's permissions or account removal generate admin events in both systems: user modified or user deleted. Endpoint and Network denial of service would not generate obvious events. In the case of a denial of service, the number of events generated compared to the type of events would be the sign of the tactic. An example would be several large exports or imports at the same time slowing down network access.

3.6. Finding Conclusions

No framework is perfect. No application is perfect. Even when applications are designed with security in mind, consideration to certain attack vectors is not reflected in available events. Good applications will minimize attack surface by removing unnecessary features and services. A detailed analysis is needed to find any gaps in coverage.

4. Recommendations and Implications

Any SaaS application needs to be reviewed independently of any self-hosted counterpart. This paper demonstrated mapping the event catalog to the MITRE Framework, while also identifying blind spots that exist. Organizations need to be cognizant of the differing nature of individual SaaS applications and understand a thorough review is a must. These are several general steps organizations can take from an engineering and governance perspective to both be aware and mitigate the risks of a migrated application.

4.1. Recommendations for Practice

4.1.1. Mapping Attack Surface

Organizations should build an evaluation process for mapping MITRE ATT&CK Tactics to the application. This should not begin with a migrated application. Best practice is to do this with every application used to understand the attack surface that currently exists. Prioritization should be given based on the impact if something happened to a given application or the data it contains. For migrated apps, the attack surface mappings should be compared to identify those blind spots that now exist.

4.1.2. Verifying Visibility

Good application developers will have an event catalog available indicating which events get logged, and what details are logged with that. It is not enough to enable logging. Those logs need to provide value. The logs can be checked against the mapping to see where visibility gaps occur. An organization needs to know its blind spots to understand what needs augmentation.

4.1.3. Adding Additional Visibility

Additional event logging should be added to create or augment coverage. This additional event logging should focus on those attack surface blind spots in the surface, e.g., the three tactics not on the Cloud SaaS Matrix (Reconnaissance, Resource Development, Command and Control). Some SaaS applications have integrations with Content Delivery Networks (CDNs) that will provide a layer of network visibility that can enable an organization to detect reconnaissance and command and control traffic. Some SaaS applications will have Software Application Programming (SAP) overlays that integrate with SaaS platforms to control add-ins and auxiliary services. These SAP overlays can be used to monitor resources, to see when an adversary may develop resources by implanting new services or add-ins.

4.1.4. Removing Attack Surface

Though a technique can affect SaaS applications in general, it does not mean controls cannot be put in place to mitigate those techniques. Brute forcing only works if the adversary has access to the login page. Most SaaS applications have Security Assertion Markup Language (SAML) integrations which allow for Single Sign On (SSO) applications like Okta. By having a master application password of extreme length and routing all other access through a SSO portal, the brute force technique is functionally mitigated. Each mapped technique should be evaluated to see if and how they can be mitigated.

4.1.5. Requiring More from Vendors

Organizations should press SaaS vendors on making application event catalogs publicly documented. Without an event catalog, mapping to techniques can only be done after running attacks against the platform to generate potential events. SaaS vendors are not keen on clients attacking the applications as that creates security events down the shared responsibility model, incurring investigation and response costs for the vendor while potentially leading to a denial of service for the vendor and other customers. The more customers demand event transparency, the more likely a vendor will add it to the application documentation.

Governance teams should press SaaS vendors on their review of the shared responsibility model, and those findings should be made available. Both the customer and vendor should understand what responsibility lies where, all the way down to the hardware. Vendors may not be keen to engage like this with a single customer. Customers can connect with other customers to press together to get SaaS vendors to be more thorough in outlining the nested risk and how it's being addressed.

4.2. Implications for Future Research

Several avenues exist for future research. There is a plethora of SaaS applications out there with different attack surfaces and event catalogs that are widely used without any comprehensive tactics' mappings done. Jira and Confluence have a comparably narrow attack surface compared to a Microsoft exchange or Dropbox.

Comprehensive research into the nested nature of the shared responsibility model is an opportunity to flesh out the true depth of the rabbit hole. Most analysis only looks one level deep. There is very little research looking at a more recursive evaluation of the shared responsibility model and how the lack of visibility and understanding creates an unquantified risk.

Over the history of computing, SaaS is relatively young, yet far younger services may start to supplant it and change the risk model: Code as a Service (CaaS) and Function as a Service (FaaS). These are usually nested in IaaS but have their own attack surfaces. MITRE has not yet caught up to these emerging technologies, and CaaS and FaaS will provide avenues for new research.

5. Conclusion

Self-hosted and SaaS applications may seem similar, but present differing security concerns based on differing responsibility models and attack surfaces. If vendors can provide SaaS solutions that let organizations offload infrastructure risk and reduce operational cost, the rate of adoption will only increase. With effort and patience, an organization can understand the new attack surface and responsibility model specific to a SaaS environment. Then organizations can secure the orange, just as they did the apple.

References

Abba Ari, A.A., Ngangmo, O.K., Titouna, C., Thiare, O., Kolyang, Mohamadou, A. and Gueroui, A.M. (2020), "Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges", Applied Computing and Informatics, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1016/j.aci.2019.11.005

Abrams, L. (2023, February 17). *Atlassian data leak caused by stolen employee credentials*. BleepingComputer. https://www.bleepingcomputer.com/news/security/atlassian-data-leak-caused-by-stolen-employee-credentials/

Atlassian. (2023, April 24). *Audit log events in Jira*. Audit log events in Jira | Administering Jira applications Data Center and Server 9.10 | Atlassian Documentation. https://confluence.atlassian.com/adminjiraserver/audit-log-events-in-jira-998879036.html

Atlassian. (2023, March 29). *Audit Log Events in Confluence*. Audit Log Events in Confluence | Confluence Data Center and Server 8.4 | Atlassian Documentation. https://confluence.atlassian.com/doc/audit-log-events-in-confluence-1005333793.html

Cervone, H.F. (2010), "An overview of virtual and cloud computing", OCLC Systems & Services: International digital library perspectives, Vol. 26 No. 3, pp. 162-165. https://doi.org/10.1108/10650751011073607

Chou, D. (2018, September 28). *Cloud service models (iaas, paas, SAAS) diagram*. Cloud Service Models (IaaS, PaaS, SaaS) Diagram. https://dachou.github.io/2018/09/28/cloud-service-models.html

Culot, G., Nassimbeni, G., Podrecca, M. and Sartor, M. (2021), "The ISO/IEC 27001 information security management standard: literature review and theory-based research agenda", The TQM Journal, Vol. 33 No. 7, pp. 76-105. https://doi.org/10.1108/TQM-09-2020-0202

Hylender, C., Langlois, P., Pinto, A., Widup, S. (2023). 2023 Data Breach Investigations Report |Verizon. https://www.verizon.com/business/resources/reports/dbir/

Jibilian, I. (2021, April 15). *The US is readying sanctions against Russia over the SolarWinds cyber attack. here's a simple explanation of how the massive hack happened and why it's such a big deal*. Business Insider. https://www.businessinsider.com/solarwinds-hack-explained-government-agencies-cyber-security-2020-12

MITRE. (2023). *Enterprise matrix*. Matrix - Enterprise | MITRE ATT&CK®. https://attack.mitre.org/matrices/enterprise/

MITRE. (2023b). *SAAS matrix*. Matrix - Enterprise | MITRE ATT&CK®. https://attack.mitre.org/matrices/enterprise/cloud/saas/

Raina, K. (2021, November 1). *Microsoft ad supply chain attack emphasizes need for Zero trust*. crowdstrike.com. https://www.crowdstrike.com/blog/microsoft-ad-supply-chain-compromise-emphasizes-need-for-zero-trust/

Srinimalan Balakrishnan Selvakumaran (Institute of Technology in Architecture, ETH Zurich, Zurich, Switzerland and Basler and Hofmann AG, Zurich, Switzerland)

Daniel Mark Hall (Institute of Construction and Infrastructure Management, ETH Zurich, Zurich, Switzerland). Journal of Facilities Management. ISSN: 1472-5967. Open Access. Article publication date: 18 January 2022. Issue publication date: 31 May 2022

Tank, D., Aggarwal, A., & Chaubey, N. (2019). Virtualization vulnerabilities, security issues, and solutions: A critical study and comparison. *International Journal of Information Technology*, *14*(2), 847–862. https://doi.org/10.1007/s41870-019-00294-x

US DNI. (2021, August 10). Kaseya VSA Supply Chain Ransomware Attack. https://www.dni.gov/files/NCSC/documents/SafeguardingOurFuture/FINAL_NCSC_Emerging%20Technologies_Factsheet_10_22_2021.pdf

Appendix A – Jira Log Event List and Count

Event Name - self-hosted Jira	Count
User added to group	333
Group added to group	320
Group created	271
Group deleted	135
Permission scheme removed from project	19
Project deleted	18
Group removed from group	17
Filter updated	17
Project roles changed	16
Project version updated	12
Project component created	11
Audit Log search performed	9
User created	7
Project version created	6
Project version released	5
Workflow updated	4
Filter created	4
Sprint deleted	4
Workflow scheme added to project	3
Filter deleted	3
User deleted	3
Permission scheme added to project	3
User updated	3
Workflow scheme removed from project	3
Project archived	2
A new field added to the screen's tab	2
ScriptRunner	2
Dashboard created	2
Board deleted	1
Dashboard updated	1
Board created	1
Project created	1

Event Name - Jira Cloud	Count
User added to group	#####
User removed from group	3,807
Permission scheme updated	902
Project roles changed	260
User created	218
Custom field created	213
User updated	161
Workflow updated	148
Field Configuration scheme updated	132
Group created	77
Permission scheme added to project	70
Workflow scheme added to project	69
Workflow created	63
Project created	59
Project updated	52
Notification scheme updated	51
Project component created	44
Workflow scheme created	40
Field Configuration scheme added to project	39
Project category changed	38
Notification scheme added to project	34
Field Configuration scheme created	34
User removed from project	33
Custom field trashed	24
Project role deleted	20
Project marked as deleted	20
Custom field deleted	17
Project deleted	17
Status created	16
Request type created	12
Issue Security scheme added to project	10
Workflow deleted	10
Workflow scheme updated	10
Request type deleted	9
Custom email channel turned on	7

Cloud Email settings created	7
Cloud email channel turned on	7
Permission scheme created	6
Workflow scheme deleted	6
Permission scheme deleted	6
Field Configuration scheme deleted	5
Notification scheme deleted	5
Project avatar changed	5
JIRA Service Desk configuration	5
Global permission added	4
Custom field updated	4
Cloud Email settings changed changed	3
Cloud email channel configuration changed changed	3
Status updated	3
Group removed from project	3
Group deleted	2
Issue Security scheme created	2
Cloud email channel turned off	2
Issue Security scheme deleted	2
Project version created	2
Deleted project restored	2
Cloud Email settings deleted	2
Project category updated	1
Plan created	1
Customer permissions changed	1
Permission scheme copied	1
Customer invited notification changed	1
Notification scheme created	1
Field Configuration scheme removed from project	1

Appendix B – Confluence Log Event List and Count

Event Name - self-hosted	Count
User Added to Group	1305
User Removed from Group	129
Group Created	69
Group Deleted	21
User Deleted	16
Userr Created	11
App Disabled	3
User Details Updated	3
App Enabled	2
Audit Log Search Performed	2
Global Settings Changed	1
JFR Recording Stopped	1
User Renamed	1
Event Name - SaaS	Count
User added to group	2,766
Space permission added	906
Space permission removed	396
Content restriction added	79
User removed from group	70
Content restriction removed	70
User created	57
Page archived	50
Space deleted	20
Space created	16
User details updated	11
User deactivated	9
Page viewed	9
Group created	9
Login	5
Space configuration updated	5

Logout	5
Page template deleted	3
Space logo uploaded	3
Global permission removed	2
Global permission added	2
User reactivated	2
Page template created	1