CHAPTER 4

Securing the Enterprise with Arc

Security As Job One

The impact of enterprise security failures is increasing as tolerance for missteps decreases. Countless breaches of consumer data have often been met with token fines and pats on the head to injured users in the form of a year or two of credit monitoring, a tepid remedy that in no way resolves the situation for those whose identity was actually misused. As failures begin to impact national security and the internal workings of large corporations though, the reaction is becoming more proportionate. The now famous SolarWinds failure to protect corporate and government consumers of its security tools is resulting in lawsuits. A November 2021 suit filed against SolarWinds board[1] by two pension funds accuses SolarWinds of failures that are patently ridiculous for a company purporting to provide IT security, such as using "solarwinds123" as a network password.

The takeaway from this and other miscreant behaviors among companies turning a blind eye to security gaps is that negligence or hoping for luck to escape the sort of risks facing corporate IT today is undeniably

[1] https://news.bloomberglaw.com/employee-benefits/solarwinds-board-sued-by-pension-funds-over-massive-cyberattack

R. Maxwell, *Azure Arc Systems Management*,
https://doi.org/10.1007/978-1-4842-9480-2_4

a foolish strategy. Companies must employ tooling not only to protect their IT landscape and data, but must also monitor their supply chain and assure that, as happened with SolarWinds, the failure of a provider cannot become a threat to the company. Can an acceptable level of threat protection be accomplished, and if so, how is Arc an aid in the process of doing so?

Security leaders are under a lot of pressure to show quick wins while knowing full well that everything they do will be heavily scrutinized and challenged, and ultimately, they will pay the price for things that are not under their control. —Yaron Levi[2]

A commonly held view that internal threats can be managed by culture or comradery has persisted despite heaps of evidence to the contrary. Late 2021, a Senior Cloud Lead at Ubiquiti was charged with wire fraud and extortion for a convoluted scheme that began with abusing his access under cover of a virtual private network [VPN] to steal his employer's private GitHub repository contents and other confidential information. An administrative account was misused not only to leak data but to adjust log retention policies and obscure suspicious activities from monitoring. He then attempted to anonymously bribe the employer to the tune of nearly $2M in Bitcoin, and when that failed, he pretended to be a whistleblower exposing Ubiquiti's lack of transparency over the breach. He was eventually tripped up by junior crook mistakes and happenstance. An Internet outage temporarily removed the VPN's cloak of his home IP address, a rather inexpensive VPN service he had paid for with his very own PayPal account. Given that Ubiquiti deals in confidential trade data, the risks posed by a breach were extreme. They admirably refused to pay

[2] www.linkedin.com/in/yaronrl/

a ransom and immediately engaged the FBI. The ultimate costs to the company are in the "billions" according to the indictment[3] and include remediation, plummeting stock value and customer goodwill.

If media accounts of the illegal actions taken are correct, the losses suffered could have been prevented. The use of a service account with shared user credentials is the first red flag. Instead, the user of the service account should first have to authenticate their identity and provide a proper credential to be able to execute actions as the service account. Treating a service account as if it were a credentialed individual is an invitation to abuse. No one should be given the keys to the kingdom without showing their ID demonstrating they are members of a role with permission to execute actions as the service account. The application of this security principle is a focus for security upgrades among major cloud providers. AWS, for instance, allows a user to assume a role[4] (such as a network administrator), but on doing so, they lose all the privileges associated with roles in their own user profile. They must officially cast off the cloak of network administrator and return to their own role to again have access to its scope of access. Azure offers managed identities and service principals that work in a similar fashion and warns in its documentation against adding a service account itself to a highly privileged group such as administrators. Managed accounts on Azure thoroughly solved the problem of direct login access since "credentials are fully managed, rotated, and protected by Azure" and "No one (including any Global admin) has access to the credentials, so they cannot be accidentally leaked by, for example, being included in code."[5] This

[3] www.justice.gov/usao-sdny/press-release/file/1452706/download

[4] https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html

[5] https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/service-accounts-managed-identities

platform strength directly benefits Azure Arc operations in several ways,[6] such as the ability to use RBAC to control access to any Arc-enabled server, secure credential storage in Azure, and the option to customize security roles to assure least privilege access.

The scope of a service or management account should be limited so that if unauthorized entry were gained through a service account credential, it would not be possible to use that credential everywhere, as appeared to have happened at Ubiquiti. Imagine your favorite spy show with the super-secret underground control center. Every elevator, hallway, and room have a monstrous steel door with an impenetrable lock. However, what good is that if they all share the same key as the front door? Break into one and the rest are meaningless. A managed identity is a way to put discrete locks on each door. Like those fluorescent stamps on your hand at a music venue, you only have access backstage if you possess both an entry ticket and the extra special one that indicates you're VIP. An Azure managed account, having access to all of your application's venues, including perhaps messaging, storage, or key management, is capable of issuing a token for access to one, multiple, or all areas of a given service depending upon the role of the requestor.

Finally, monitoring of access should be continuous, and policies must be centralized in an area that has its own management credentials (not shared with the assets being governed). This is the sort of governance Arc intends to simplify, and similar tools are available on AWS.

The case of Terry Childs, who as the Senior Network Administrator locked the City of San Francisco and its collaborators out of their own network in 2008,[7] is another infamous example of the dangers of ignoring

[6] https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/hybrid/arc-enabled-servers/eslz-identity-and-access-management

[7] www.wired.com/2008/07/sf-city-charged/

internal risk. An interesting article by Paul Venezia,[8] Senior Editor at InfoWorld, implies his behavior might have been a case of someone going digitally postal rather than greed as in the Ubiquiti hack. Nevertheless, it heavily damaged the City's reputation for information governance and created a major scandal that made headlines nationwide. As in the prior case, known principles of IT security could have been applied to make sure that one person was not the sole repository of critical infrastructure access. Across the IT admin community, there tends to be a high level of professionalism, integrity, and pride in what is viewed by most as a sacred trust in their ability to protect the interests of the enterprise IT landscape they are entrusted with. Nonetheless, humans are a big variable in the overall security posture of an organization – which leads us to another common misconception – or misnomer, the *human firewall*.

The late Kevin Mitnick[9] was a security expert who was imprisoned twice in the late 1980s and early 1990s for his adeptness at breaking into corporate IT systems partnered with Stu Sjouwerman in building KnowBe4, a consulting firm devoted to showing the emperors of IT security exactly where they can find their clothes. His classic social engineering techniques were demonstrated by KnowBe4's Roger Grimes in a presentation on the weaknesses of multifactor authentication [MFA], a strategy widely regarded as an effective way to block account impersonation. Armed with only a user's name, phone number, and email, he was able to successfully convince callers to provide enough credential information to intrude into organizations that thought they had provided sufficient training to protect against such attacks. In fact, the National Institute of Standards and Technology's Digital Identity Guidelines[10] have degraded SMS push notices to "out-of-band second authentication

[8] www.infoworld.com/article/2653004/why-san-francisco-s-network-admin-went-rogue.html

[9] https://gizmodo.com/kevin-mitnick-famous-hacker-dies-at-59-1850659160

[10] https://pages.nist.gov/800-63-3/

factor"[11] for the reason that it's difficult to validate that the SMS is actually going to a cellular phone in the user's possession vs. a VOIP number that may have multiple access points across any PC which the user has logged in to.

Since the human firewall has as many vulnerabilities as there are types of humans on which to practice social engineering techniques, this brings us back to automated monitoring and threat response to shutter intrusion attempts before major damage can be accomplished. Historically, there has been resistance to efforts to lock the inner doors. A decade ago, many administrators did not want to deal with implementing HTTPS to encrypt internal traffic, nor were CFOs willing to finance it – now it is a baseline standard that comes baked into the majority of infrastructure and application traffic. The Secure Access Service Edge [SASE[12]] model is now facing the same introductory pressure, but is fast becoming a standard among cloud providers and large enterprises. In practical terms, it can be thought of as integrating security into every single artifact of an IT ecosystem from a global perimeter to a tiny IoT device on the factory floor. It is the technological manifestation of the Home Alone hero Kevin's elaborate defense strategies.[13] When that pernicious nation-state burglar gleefully assumes that they're inside your infrastructure, there will be a paint can smack to the head waiting around the next corner.

As of early 2024, a retrospective of significant cyberattacks in 2023 doesn't show a reversal in the level of risk; in fact, annual reports issued by IBM,[14] Fortinet,[15] and Akamai[16] each demonstrate that attack vectors

[11] www.nist.gov/blogs/cybersecurity-insights/questionsand-buzz-surrounding-draft-nist-special-publication-800-63-3

[12] www.zscaler.com/resources/security-terms-glossary/what-is-sase

[13] www.imdb.com/title/tt0099785/

[14] www.ibm.com/downloads/cas/DB4GL8YM *and* www.ibm.com/downloads/cas/E3G5JMBP

[15] www.fortinet.com/resources-campaign/cloud/2023-cloud-security-report

[16] www.akamai.com/resources/state-of-the-internet/2023-year-review

and methodologies continue to expand apace with the growth of technology and that AI has become significantly more important to both cybercriminals and organizations seeking to defend against them. Both IBM and Akamai highlight IoT devices as an inherently insecure area to which special attention must be given, and Akamai cites CSO[17] in calling out insufficiently managed Internet of Medical Things [IoMT] as "some of the most vulnerable assets across all industries." IBM focused notably on the costs of a data breach reporting that it had escalated more than 50% from 2020 to its Q1 2023 assessment and that companies who neglected to involve law enforcement or discover the breach themselves were likely to incur substantially larger recovery costs. Fortinet, providers of a multi-cloud security platform, noted that more than two-thirds of their survey respondents operate on more than one public cloud and that even when facing "macroeconomic headwinds" a majority of customers are increasing their security spend. Later chapters of this book focusing on monitoring, policy, and automation capabilities enabled by Azure Arc will discuss how its control plane offers visibility into IoT workloads and integrates with security offerings capable of diminishing many of these threats.

Monitoring – Light in the Corners of the IT Universe

If one consistent message echoes through the daily barrage of security incident and breach reports, it is that danger is constantly lurking. It is also true that the volume of data most organizations must monitor is beyond human capabilities to sift through. Traditional monitoring solutions which offer reports after the fact are of low value if the horse is already out of the barn, or worse yet the wolf is inside selecting its next target. To be effective,

[17] www.csoonline.com/article/651075/new-research-reveals-most-attacked-vulnerable-assets.html

monitoring needs to be as near real time as is physically possible and paired with an effective threat response. Thus, in many cases, monitoring a log of what *has* occurred will be ineffective, and what is needed is to respond to events in real time. Then the response to the event can be altered to protect the system being monitored. The log, meanwhile, will contain forensic data that can be used to harden that same system against future attacks.

Arc's approach to monitoring benefits greatly from the product having its genesis in Azure. For security monitoring, Arc-enabled servers can be connected to Microsoft Sentinel,[18] a premier SIEM/SOAR[19] tool that was ranked at the top of Security Operations tools by Gartner in 2022's Magic Quadrant for that category.[20] Meanwhile, container workloads benefit from Container Insights which captures actionable performance data. We will examine usage scenarios for both of these products as part of a discussion of process automation in Chapter 8.

If you look at the InfoSec Institute's eight domains risk[21] monitoring may technically belong to Security Operations, but in practical terms, it's a nonfunctional requirement of all of them. Arc lands squarely in the Security Operations domain with its emphasis on server management for purposes of patching, application of policy, and continuous monitoring.

Integration with Lighthouse

Lighthouse is an interesting offering from Microsoft because it doesn't have any true competition from the other two major cloud vendors. It's an integrated solution to allow companies that run environments for

[18] https://learn.microsoft.com/en-us/azure/sentinel/overview

[19] https://www.paloaltonetworks.com/cyberpedia/what-is-soar-vs-siem

[20] https://techcommunity.microsoft.com/t5/microsoft-sentinel-blog/microsoft-is-named-a-leader-in-the-2022-gartner-magic-quadrant/ba-p/3666566

[21] https://resources.infosecinstitute.com/certification/the-cissp-cbk-domains-info-and-updates/

thousands of customers to centrally manage them. For example, huge consulting companies like EY whose suite of tax applications serve legions of tax clients, each discretely separated into their own application space,[22] could benefit from Lighthouse. Lighthouse harkens back to Microsoft's SharePoint roots and expertise in secure multi-tenancy and puts that on steroids with cloud solutions running on Azure. As the documentation for Lighthouse explains, "Authorized users, groups, and service principals can work directly in the context of a customer subscription without having an account in that customer's Azure Active Directory (formerly Azure AD, now Entra) tenant or being a co-owner of the customer's tenant."

Whether the scope being managed in the customer's tenant is the entire subscription or just a resource group, the customer's subscription ID is still required to complete registration for the assets that will be administered. Registration also requires the managing organization be assigned to roles that have RBAC to `/register/action` in the client's tenant, whether that be as a contributor or owner in the client's tenant or automated by use of a Logic App that has the appropriate permission.[23] The managing organization does not work directly in the client's tenant; rather, Lighthouse creates a shadow copy of the assets under management. When an administrator in the managing organization makes a change to their copy of the tenant, Lighthouse checks the registration and permissions before writing the change to the actual tenant in the client's organization. All activities and the identity they were performed under are also logged in the client's tenant, and the *client tenant can remove access*.[24] This is an important point to bear in mind when designing a service

[22] www.ey.com/en_us/tax/global-tax-platform (one example, see video for clarity)
[23] https://learn.microsoft.com/en-us/azure/lighthouse/how-to/onboard-management-group#register-the-resource-provider-across-subscriptions
[24] https://docs.microsoft.com/en-us/azure/lighthouse/concepts/architecture

portal. Are you delivering a product where the administering organization also owns all of the portals being administered? Or simply delivering an administrative service?

How does this dovetail with Arc, and in what way does Lighthouse combined with Arc enhance the ability to monitor and secure the IT landscape? We've been discussing Arc as though it lives at the pinnacle of the administration pyramid, but when managing tenants with Lighthouse, it is actually the latter in the top perch. In addition to the aforementioned scenario of an organization with many clients consuming a SaaS application, this overview also has great value in enterprises with a large number of subscriptions. While Lighthouse does not operate at the Management Group [MG] scope, a policy for the MG can specify that all of its subscriptions be registered with Lighthouse, thus gathering a vast IT landscape into a single high-level view. With Lighthouse, you can peer into Arc instances in all the managed tenants you operate for clients as well as the vast expanse of resources you utilize internally and thus extend Arc's ability to consistently manage the server's and Kubernetes clusters across your continually expanding IT galaxies.[25] Yes, go ahead and pick up your scepter. With that kind of power, you are truly a ruler of your IT-verse.

Private Link

The value of log files is often underestimated given their pedantic recitation of every routine operation of the system or application emitting them. In truth, a log might store trade secrets or competitive opportunities, security risks, or intrusions against the same, performance and reliability data, and so much more. Thus, the logs themselves become a target for theft and malfeasance.

[25] https://docs.microsoft.com/en-us/azure/lighthouse/concepts/cross-tenant-management-experience#enhanced-services-and-scenarios

The idea behind Private Link is that none of your log traffic will travel on the public Internet. Instead, it will travel on Azure's internal backbone, and when it crosses a boundary outside of that network, it will run through a protected tunnel everywhere else (e.g., ExpressRoute). This sounds ideal and combined with encryption is a near-impenetrable solution; however, there are some caveats.

A severe constraint on any solution, not only those pertaining to security, is that it must integrate with existing systems. Microsoft describes Private Link as "constellation of different interconnected services that work together to monitor your workloads." To translate that elegant statement to English, Private Link has been stapled on to Azure in order to provide much needed security enhancement. The impact to an organization wishing to implement Private Link is that an entire existing network topology may be affected.

Microsoft describes the impact as "setting up a Private Link even for a single resource changes the DNS configuration affecting traffic to **all resources**. In other words, traffic to all workspaces or components are affected by a single Private Link setup." Their networking guide for Private link further elaborates, "Some networks are composed of multiple VNets or other connected networks. If these networks share the same DNS, setting up a Private Link on any of them would update the DNS and affect traffic across all networks."[26] Microsoft's suggested resolution is that there should be only one Private Link scope in all of an organization's shared DNS. Clearly, the implementation of Private Link should be treated as a potentially breaking change if there is existing shared DNS.[27] It is worth noting that if you are a commercial or government client, Private Link can be utilized not only in conjunction with monitoring but also as a private

[26] https://docs.microsoft.com/en-us/azure/azure-monitor/logs/private-link-design#plan-by-network-topology

[27] www.networkworld.com/article/3268449/what-is-dns-and-how-does-it-work.html

route for Azure Automation. This can offer a greater return on the effort to set up Private Link as will be explained in a later section of the book covering process automation.

Security

Secure Access Service Edge (SASE)

The need for new ways to look at security has been underscored over the past decade's trickle and then rush to the cloud, along with the situation of the very scenario of hybrid and cross-cloud distributed computing that Arc is designed to address. The standard paradigm of establishing a traffic route to the data center, guarding the entrance with a firewall, and piping traffic in and out through a VPN simply doesn't cover the yawning gaps that new infrastructure and application models have opened. For instance, the traditional monolithic app sitting within the data center has changed to massive proliferations of APIs delivering services, each of which needs to be examined from numerous perspectives from the beginning of the application lifecycle when determining the scope of what the API will expose to monitoring queries against it when live. And APIs are just one segment of the "Service Edge" around which "Secure Access" must be constructed.

SASE inverts the traditional model of siloed and sometimes disparate authorization and authentication models by moving those functions out to the perimeter or edge of the computing environment. Companies such as Cloudflare, AWS, Microsoft, Akamai, and others maintain global networks that form an "edge" or perimeter through which all traffic must pass. What this facilitates is the ability to examine requests while they are far away from the actual resources you are protecting. The identity of the requestor can be authenticated and then impersonation attempts blocked by examining the context and other risk signals at the moment of the interaction.

We have already touched on ways Arc facilitates edge computing scenarios, and SASE implementation is implicit in some of what has been discussed in terms of securing individual assets so that request traffic can only travel on paths where specific access has been granted (e.g., permission to the results of a specific query but not to its data source). To zoom out to the big picture though, SASE enhances security by simplifying traffic patterns and management. SASE is an architectural paradigm that may dictate a change to network traffic patterns. Instead of protecting each co-location with its own authentication providers and authorization controls, those functions move up to the cloud, and a key benefit is that an entire distributed architecture can benefit from a single sign-on whether accessing corporate systems, retail locations, edge compute functions, or any other asset.

In addition to industry standard security and compliance frameworks that Azure complies with, it has its own set of standards, the Microsoft Cloud Security Benchmark.[28] The benchmark attempts to address a majority of the functions and services available on Azure, as well as external systems managed using Arc, and map each one to security controls that are "consistent with well-known security benchmarks, such as those described by the Center for Internet Security (CIS) Controls, National Institute of Standards and Technology (NIST), and Payment Card Industry Data Security Standard (PCI-DSS)."[29] This very granular implementation guide can serve as an invaluable security checklist that will ideally be consulted from planning stages forward. The responsibility for implementation will vary with architecture, since some Azure service offerings will remove the need to manually construct security.

A current challenge in SASE implementation, or even distributed applications in general, is performance. SASE vendors heavily compete on this aspect, with Cloudflare currently claiming best performance and

[28] https://docs.microsoft.com/en-us/security/benchmark/azure/overview

[29] www.cisecurity.org/cis-benchmarks

Akamai asserting they have more edge locations than any other vendor. Microsoft doesn't limit its customers to their own Front Door offering, but facilitates integration with all of the major vendors (as is the case with AWS and GCP also).

Since the human firewall will always have limitations, and frictionless access to key business resources is key to business operations, building security into every layer of your infrastructure should be a primary objective. Criminals seeking entry into your company's systems are very thoroughly assessing every possible attack vector, and intrusion attempts are often automated so that waiting for a human response to discover and respond to a threat would be a futile exercise. In a future chapter, we will discuss how Arc's monitoring feedback loop can be closed by an automated response to identified threats.

Role-Based Access Control (RBAC)

Role-based access control [RBAC] is not a new concept in IT security. According to the National Institute of Standards and Technology [NIST],[30] instances of RBAC can be found as far back as the 1970s or well before the advent of personal computing. NIST also notes a key point that RBAC differs from simple security groups in that members of a group may have individual security permissions assigned to them, whereas in a true role-based security schema, roles hold only activities, and thus a user or group may only perform an activity if they are assigned to a role which has permissions to perform it. Groups still exist for the aggregation of users, as it is simply not practical to provision thousands of individuals manually, but for RBAC to function correctly, it is important to focus on the roles membership provides.

[30] https://csrc.nist.gov/projects/role-based-access-control/faqs

A classic permission problem has been the difficulty of assuring a user who changes to a new position in an organization sheds the privileges belonging to their former assignment. Over time and the user's career advancement, this can lead to organization-wide privileges, a situation diametrically opposed to another key security principle that users have the least privilege required to accomplish their tasks.[31] In order to use RBAC effectively then, a role change must be treated as a new hire, and prior role assignments must be removed before creating new ones. Additionally, the level of privilege a user should have does not necessarily correlate to their authority in the organization since seniority within a company may reflect on a person's business acumen rather than their technical proficiency. Thus, an executive may have few limits on information access, but severe restrictions on the ability to unhook policy controls or change infrastructure – all in the interests of their own and the company's protection from liability.

Today's authentication and authorization tools are many leagues better than those available even a decade ago. Today, Microsoft Entra[32] (formerly Azure Active Directory) can effectively manage business-to-business [B2B] relationships and allow partners access to resources internal to your organization without allowing them to federate their Active Directory with yours or requiring you to create an account in your organization for them. This can be particularly useful in terms of vendor management so that, for instance, application developers can be given access to development environments but be completely restricted from areas like quality assurance or production environments for the safety of the company's internal data. You can also put additional authentication controls on B2B accounts, such as requiring an additional multifactor authentication step (e.g., entering a code received via text message or through an

[31] www.onelogin.com/learn/least-privilege-polp

[32] https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/custom-overview

authentication application) which can be particularly valuable if their home organization's security controls do not meet internal standards. A particularly egregious example of that was the infamous Target breach of 2013, wherein an HVAC contractor fell victim to a phishing scam and their access to Target's vendor systems (which were not well protected in terms of network segmentation from the rest of Target's infrastructure) was misused by hackers to install software to steal customer's credit card data directly from point-of-sale systems at checkout.[33] That HVAC vendor was using a free version of an anti-virus product that did not perform real-time detection and hence did not detect the attack. Target paid some of the highest regulatory fines ever assessed to that point in time for compromising PID (personally identifying information) data as punishment for lax security controls.

Entra also allows you to time-box authorization to resources for any user, internal or external, for periods as brief as five minutes.[34] You might think it would be advantageous to frequently have users reenter passwords, but many security experts do not favor this approach since it offers more opportunities for users to be compromised while entering their credentials and is not particularly valuable on a device managed by the organization. However, short sessions are of real value when working with external partners to assure they frequently validate their membership in the organization you have contracted for services.

Entra also covers business-to-client [B2C] scenarios elegantly so that users of an application hosted on Azure may use local or external authentication providers such as Facebook or Google to log in to the application (while having zero access to anything else in your organization).

---

[33] www.commerce.senate.gov/services/files/24d3c229-4f2f-405d-b8db-a3a67f183883

[34] https://learn.microsoft.com/en-us/entra/id-governance/privileged-identity-management/pim-how-to-add-role-to-user

With the previous examples of security failures in mind, it becomes apparent that security personnel should be viewed as "special forces" among the "citizen army" protecting your organization. Hiring and retaining talent that understands the modern security landscape along with promoting a culture of integrity throughout the organization is an indispensable aspect of IT security for the business. Further, it's important to minimize the destructive impact that any one failure in the human fabric of the organization can accomplish.

In the space of server management, a few individuals may manage very large portions of the IT landscape. Even if there is proper network segmentation and other controls protecting against external threats when an administrative group has permission to cross those boundaries, there is inherent risk that can only be mitigated somewhat by centralized management and appropriate monitoring.

Azure has comprehensive server management solutions that Arc is designed to extend beyond the boundaries of the Azure platform. This is a special boon in terms of security for large distributed server installations which can now be managed as Azure resources. For actively connected servers, it allows you to use Azure RBAC across all of your organization's assets, be that Kubernetes clusters living in AWS, web servers running Linux, or on-premise SQL Servers in your data center. All of these can have the same access policies and role assignments. Local accounts on servers need only be used to onboard the server to Arc initially by installing the agent, and from that point forward, the agent serves as proxy to execute administrative actions on the machine that is under Arc's purview. If there were only one "killer app" feature that stands out as justification for the cost of running Arc in an organization, the ability to unify server security administration under Arc's umbrella might be it.

Security Risks Resulting from Arc

While Arc can contribute greatly to the security posture of an organization, it's worth noting that Arc itself can be an attack vector. The security principal used by Arc agents could be compromised as described in a June 2021 blog by Matt Felton.[35] While, as Felton concludes in his article, this security risk is a trade-off that is "leagues better" than common industry practice, it is not one to be ignored. Organizations implementing Arc will want to apply appropriate security controls and monitoring to this portion of the infrastructure as any other, assuring that the service account running an agent has only required privilege for the jobs it is assigned and that credentials for these accounts are rotated regularly.[36]

In truth, the risks highlighted for Arc's agents are common to using agents on a server in general for tasks like virus scanning, backup, and more. There are also management issues such as the risk of missing or gaps due to agent incompatibility with various operating systems and legacy hardware, while agentless approaches such as using network scans may miss disconnected devices. As management platforms have become a requirement to run operations at scale, I think you can expect the development of new protocols to facilitate running administrative controls in a secure and friction-free manner.

Myriad Risk Factors Require Thoughtful Design

In terms of security, changes to application protocols have also posed new challenges. Cloud-native development relies heavily on APIs to interact with application models, and ways of exposing those interfaces

[35] https://journeyofthegeek.com/2021/06/12/experimenting-with-azure-arc/

[36] https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/service-accounts-governing-azure

vary from classic REST to modern GraphQL. As the move to cloud-hosted applications has accelerated, the popularity of APIs as an attack vector has also grown. An F5 Labs security article notes, "some of the largest and well-known companies—Facebook, Google, Equifax, Instagram, T-Mobile, Panera Bread, Uber, Verizon, and others—have suffered significant data breaches as a result of API attacks."[37] One Gartner webinar[38] cited research indicating that organizations using an external API management solution had more robust security profiles in terms of API protection, highlighting the value of objectivity in assessing risk profiles.

API protection starts left at the design phase of the application lifecycle as discussed earlier in this book. Application design must carefully consider exactly what information is required from the data source behind the API and strictly limit the ability of the application to make API calls to only the required scope of data (avoiding what OWASP calls "Excessive Data Exposure"[39]). Consideration must also be given to how even that limited access could be misused by manipulation of the query language used, and when the API is under development, it should be evaluated for security flaws of this nature. Protection of the data source itself is also key, and this highlights the importance of data stores being an integral part of the DevOps pipeline. There are many ways to surface data for application use without allowing an API access to the actual database. Applications should be designed to enumerate queries, so that simple replay attacks cannot succeed. Finally, carelessness or malfeasance can put APIs at risk when encryption keys and other secrets are stored in the codebase. It is

[37] www.f5.com/labs/articles/education/securing-apis--10-best-practices-for-keeping-your-data-and-infra see footnotes directly in article for company scenarios listed

[38] www.gartner.com/en/webinars/4002323/api-security-protect-your-apis-from-attacks-and-data-breaches

[39] https://owasp.org/www-project-api-security/

helpful to remember that a single application may draw and update data from or even circulate it among multiple APIs, thereby broadening the attack surface in even a small application.

If left solely to an internal development team, security could be compromised by delivery pressure, individual skill profiles, and even a lack of corporate enthusiasm for updating software where later versions benefit from updated security. No organization should ever consider their applications to be protected without a deliberate security strategy (the "Sec" in DevSecOps) that can provide continuous feedback to the development team creating an application. Simply discovering the number of APIs present can be a significant challenge, since API calls are part of the core development paradigm for many development frameworks. Scanning for internal and externally facing APIs so that they can be registered with an API protection platform is the critical mitigation step to assure that undocumented APIs do not put the organization at risk.

What processes exist in Azure for API protection, and how do they integrate with Arc? Azure's API Management service offers full lifecycle management of APIs and includes the ability to apply policies, perform API discovery, manage API gateways, and more.[40] Azure also offers an API gateway within Azure as a managed service and the ability to use a self-managed API gateway on non-Microsoft and other on-premise API hosts.[41]

A gateway provides many features that enhance API security, but should not be considered a stand-alone security solution. Advantages include avoiding direct connections to API endpoints (instead, the gateway provides a proxy connection), HTTPS traffic, and certificate management. Gateways can also accomplish request routing to numerous APIs to improve an application's responsiveness, sometimes reducing the complexity of the application itself by offloading these features onto the

[40] https://azure.microsoft.com/en-us/services/api-management/#overview

[41] https://docs.microsoft.com/en-us/azure/api-management/self-hosted-gateway-overview

gateway. Gateways can also manage authentication to the application and improve performance with cached response data. However, the gateway alone is not a comprehensive API security solution and must be combined with policies governing API usage, targeted monitoring, and appropriate safeguards against abuse when it is detected.

An API gateway is not to be thought of as a singular effort. Multiple gateways prevent performance bottlenecks and are a standard in large organizations. For a self-hosted gateway, it will be important to link distributed systems to Arc management controls, for instance, Azure Monitor is not automatically available in a self-hosted gateway, but the gateway can be configured to export its logs so that they can be consumed by Azure Monitor.

Arc provides the ability to apply policies to protect APIs across a hybrid server landscape. A few examples of the type of policies that might be employed for API protection include tracking where a request to the API originates and blocking suspicious origins, limiting the rate and number of requests allowed, rejecting requests where the data contained in the response does not match predefined parameters, as well as specifying what type of authentication is required and the length of time an authorization for data will exist. Additionally, when using a gateway, policies can be used to assure internal application URLs are never exposed, but are instead replaced by gateway URLs. With the ability to deploy some Azure services through Arc-enabled Kubernetes cluster extensions, you can now run Azure's API Management Gateway on clusters in your own DC[42] or on a competitor's cloud (in preview as of this writing). For companies with many API-first workloads, that feature alone is likely to motivate adoption of Arc.

In summary, neither this chapter nor this book is sufficient to cover every aspect of enterprise security controls, but can hopefully highlight key areas on which to focus attention and the need for rigorous attention to security management. Next, we'll take a look at Arc-enabled data services.

[42] https://learn.microsoft.com/en-us/azure/api-management/how-to-deploy-self-hosted-gateway-azure-arc