# Behind the Scenes: Securing the Infrastructure Powering the Microsoft 365 Service

Art Sadovsky, Raji Dani, Chang Kawaguchi, Joel Hendrickson

## Contents



## Introduction

Microsoft 365 is one of the world's largest enterprise and consumer cloud services, and customer trust is the foundation of our business: organizations across the world rely on us to securely handle and maintain their most critical information assets across e-mail, documents, and other communications. To maintain that trust, we invest heavily in securing the infrastructure that powers our services and hosts this data on behalf of our customers – keeping customer data private and secure is the top priority for our business.

The service is growing rapidly, both in terms of our customer base and in terms of the products and experiences we provide to our customers. Meanwhile, attacker groups seeking to exploit enterprise and consumer data continue to evolve. Customers looking to secure their most sensitive data are going up against the most sophisticated and well-funded adversarial organizations in the world, including nation state attackers with seemingly limitless resources. To secure their data in the face of this already substantial and constantly evolving attacker landscape, while continuing to benefit from world-class productivity solutions and experiences, customers cannot settle for anything less than the most advanced

security solutions. This paper will explain how the work we do to secure the Microsoft 365 service infrastructure is a critical part of how customers can achieve the protection they need to stay ahead of the world's most challenging and dangerous threats.

Our goal with this paper is to shed light on how we secure one of the world's largest enterprise and consumer services. Much of this work is "invisible" to our customers. When we manage the service infrastructure and operations on their behalf, the work we do to build and maintain that infrastructure securely is not directly observable by end users of the service. But we want to give you a look behind the scenes and demonstrate how much security value customers inherit by trusting Microsoft 365 services with their data.

We will focus on the following questions:

1. What are the key principles underlying our approach to service security?
2. How do we secure our service infrastructure and the customer data it contains at the scale of a complex, worldwide offering (indeed, how do we use that scale to our advantage)?
3. How do we maintain our security posture as new services are added and existing services expand with new functionality, resulting in greater attack surface area?
4. How do we stay ahead of attackers, so that our infrastructure is resilient in the face of an evolving attacker landscape?

A single paper cannot capture all the security work done as part of Microsoft 365. This paper will focus on the key principles and work done to secure our infrastructure, and we will build on that with additional papers deeply covering specific security topics. These topics will range from the security implications of AI and intelligent products, to guidance we offer customers for securing their assets using Microsoft's customer facing security products. By articulating the principles behind service security, this paper will provide a foundation for the rest of this series.

## What is Microsoft 365?

Microsoft 365 is the cloud-powered, subscription-based version of Office, Windows 10, Enterprise Mobility + Security, and Compliance. Microsoft 365 customers get clients such as Outlook and Windows, and they also benefit from services that Microsoft hosts on their behalf, such as Exchange Online, Microsoft Teams and SharePoint Online. All components of the service are regularly updated as part of the subscription model, so that our customers have an "evergreen" product. Microsoft manages the service infrastructure on behalf of customers, meaning that Microsoft is responsible for securing the infrastructure that stores customer data.

In terms of scale, we currently use close to a million machines to power Microsoft 365 services. The infrastructure powering these services varies widely across service-specific hardware and virtualized environments in Azure, Windows and Linux, and multitenant and dedicated platforms. Microsoft 365 is a global business and our infrastructure is distributed in datacenters around the world, enabling our customers to meet data residency and sovereignty requirements.

In short, the service is complex and runs at incredible scale, and it requires thousands of Microsoft engineers to build and maintain. It is our top priority for us to keep all this infrastructure secure.

## Delivering on the promise

In the next several sections, we will look at our major investment areas for securely building, running and maintaining the Microsoft 365 services:

- Building tools and architecture that *protect* the service from compromise
- Building the capability to *detect and respond* to threats if a successful attack does occur, such that we can recover service as quickly as possible with minimal adverse consequences
- Continuous *assessment and validation* of the security posture of the service, such that we can identify potential threats to the service before they materialize

Observant readers will notice that this closely maps to the commonly used NIST cybersecurity framework. We find this to be a valuable way of thinking about the categories of work needed to run a service securely at scale. In fact, we see security and compliance as deeply interrelated topics, as the work we do to secure our infrastructure naturally accrues to our ability to meet compliance and regulatory standards such as NIST. We'll explore the relationship between security and compliance, and the compliance benefits our customers inherit by using the service, in future papers.

### Designing for Security: Core Principles

Before getting into details, we wanted to describe the major principles that influence our approach to service security:

- **Data Privacy:** We strongly believe customers own their data, and that we are just custodians of the service that hosts their data. Our service is architected to enable our engineers to operate it without ever touching customer data unless and until specifically requested by the customer.
- **Assume Breach:** Every entity in the service, whether it is personnel administering the service or the service infrastructure itself, is treated as though compromise is a real possibility. Policies governing access to the service are designed with this principle in mind, as is our approach to defense in depth with continuous monitoring and validation.
- **Least Privilege:** Access to a resource is granted only as needed and with the minimal permissions necessary to perform the task that is needed.
- **Breach Boundaries**: The service is designed with breach boundaries, meaning that identities and infrastructure in one boundary are isolated from resources in other boundaries. Compromise of one boundary should not lead to compromise of others.
- **Service Fabric Integrated Security:** Security priorities and requirements are built into the design of new features and capabilities, ensuring that our strong security posture scales with the service. At the scale and complexity of Microsoft 365, security is not something that can be bolted on to the service at the end.
- **Automated and Automatic:** We focus on developing durable products and architectures that can intelligently and automatically enforce service security while giving our engineers the power to safely manage response to security threats at scale. Again, the scale of Microsoft 365 is a key consideration here as our security solutions must handle millions of machines and thousands of internal users.
- **Adaptive Security:** Our security capabilities adapt to and are enhanced by continuous evaluation of the threats facing the service. In some cases, our systems adapt automatically through machine learning models that categorize normal behavior (as opposed to attacker behavior which would

represent a deviation from norm). In other cases, we regularly assess service security posture through penetration testing and automated assessment, feeding the results of that back into product development.

The next sections will describe how these principles come to life within the framework of *protecting* the service from threats, *detecting and responding* to any threats that do materialize, and continuously *assessing* the security posture of the service and improving that based on the results of those assessments.

## Service Protection: Minimizing the Risk of Compromise

Broadly speaking, service protection focuses on two vectors: people (making sure that Microsoft employees who build and manage the service cannot compromise or damage it), and the technical infrastructure of the service itself (making sure that machinery running the service has integrated defenses and is architected and configured in a most-secure default configuration).

### Defending Against Insider Threats

Our motto here is Zero Standing Access (ZSA). This means that, by default, the teams and personnel charged with developing, maintaining, and repairing core Microsoft 365 services have no elevated access to the service infrastructure. By default, these users only have a basic account which cannot be used to access data or perform any invasive actions. Any requests for elevated access are closely scrutinized, scoped to a minimal subset of the service, require layers of approval, and only take effect for a limited time. This is our application of the assume breach mentality to our internal employee accounts: while we perform strong background checks on our engineers, we do not assume that they are trusted when it comes to operating the service. An account could be compromised, or a user could go rogue. Our approach to access control assumes this possibility and builds in protections against it.

ZSA and running a service with least privilege is a tall claim, and achieving that state requires an architecture meticulously designed with this principle in mind. Under this architecture, the operations required to run the service can be thought of as discrete tasks, with millions of these tasks running each day across the service. Some of these are automated while others are triggered manually by an operator, but in all cases each task requires explicit permission to act against a target resource – and this permission is granted by a one-time use token bound to the specific task and target. This scoped token allows us to maintain access isolation, including the ability to ability to isolate sovereign environments. The architecture making all this possible is known as *task-based access control.*

#### Task-Based Access Control: A Closer Look

Ultimately, task-based access control is meant to enable our engineers and systems to safely take only those actions that are needed, and only where they are needed, without the need to maintain highly privileged accounts that have standing access to perform a broad range of administrative actions. This is a direct consequence of the assume breach mentality underlying our approach to service security: the assumption that any account is at risk of compromise compels us to design systems in a way that minimizes the blast radius of those accounts.

It's important to note that every engineer operating in the datacenter must pass regular mandatory background check and an annual security and privacy training to hold a valid account (and this is just to maintain a basic account that cannot, by default, perform sensitive operations!). Once engineers complete this background check, they go through a rigorous multi-factor authentication system that verifies

multiple aspects of their identity before enabling this basic account. We maintain real time identity management systems to enforce this – if an engineer is ever found to be non-compliant (for example, through a lapse in training or through leaving the company), the account is disabled immediately.

The basic account described above is distinct from the account that an engineer uses to access corporate resources like e-mail and does not grant direct access to the production service infrastructure. The basic account simply gives our engineers the ability to request elevated access if they have a legitimate business need to do so. This access is, in turn, governed by the principle of *least privilege*. This means that all access is granted within the boundaries of Just-in-Time (JIT), and Just-Enough-Access (JEA) policies. That is, elevated access is only granted at the time of need and only for a fixed duration. Moreover, elevated access only gives the user the ability to perform the specific actions they need, and for a specific scope of service infrastructure. We do not allow global administrative access. All of this is implemented through a system we call Lockbox – it ensures that any user requests for elevated access get enough approval and are then implemented with JIT and JEA controls.

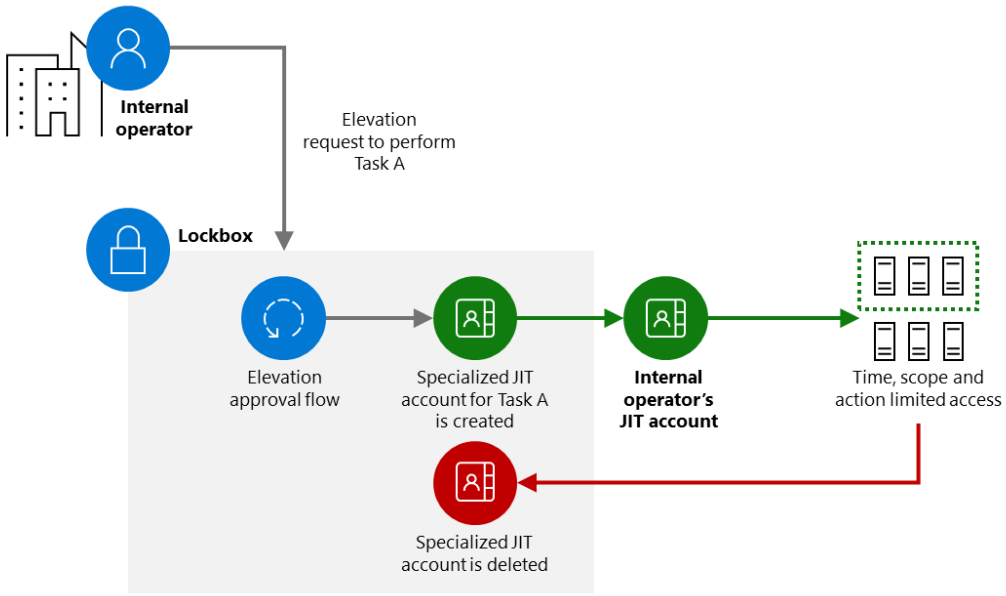


*Figure 1: Illustration of the Lockbox JIT request process. No account has standing administrative rights in the service. Just in time (JIT) accounts are provisioned with just enough access (JEA) to perform the action that is needed*

Rather than having users interactively log into service infrastructure directly, we have implemented tools and workflows that securely perform most of the actions needed to troubleshoot or recover from an issue with the infrastructure. These tools are made available to the account once JIT and JEA approval have been granted. For those rare instances when direct access to infrastructure is needed, users are required to obtain additional levels of approval and must use a Secure Access Workstation (SAW) to perform that action. And, of course, we have detailed auditing and monitoring for all this activity.

Again, a major assumption driving our design and management of the service is that any account can be compromised. Our enforcement of JIT, JEA, and task-based access control using Lockbox ensure that the potential damage resulting from a compromise can be minimized and detected. Consistent with our

security principles outlined earlier, the entire Lockbox system is deeply integrated into the service: as we scale the service both horizontally and vertically, Lockbox automatically scales as well, ensuring that our access control mechanisms always apply throughout the service.

## Encryption and Secrets Management

Securing privileged access to our services through modern access control is a critical part of protecting the service, but it is not enough by itself. We also defend the confidentiality and privacy of our customers' data as it moves from the clients to the service, in addition to enforcing strong privacy controls on the data that gets stored in the service. We achieve this through encryption throughout the data lifecycle, meaning that we encrypt both data in transit and data at rest. This is an example of some of our key principles such data privacy and assume breach at work. Nothing is taken for granted when it comes to keeping customer data secure and private.

### Encryption for data in transit

Data in transit can be defined as one of the following:

- Communication between a client machine and Microsoft 365 services
- Communication between two Microsoft 365 servers in different datacenters
- Communication between a Microsoft 365 server and a non-Microsoft 365 server

All of the above takes place over encrypted protocols such as TLS. Our services prefer the most secure ciphers and protocols, enabling customers with modern clients to get the best possible assurance of privacy in transit. We regularly reassess the configuration we've chosen as new threats emerge and improved protocols become available. We onboard these newer protocols into our services and deprecate older and weaker protocols methodically and transparently. This ensures that communication with the service, whether it is inter-datacenter between Microsoft 365 servers or customer facing, is protected with strong encryption.

### Encryption for data at rest

Protecting customer data at rest with strong encryption is one of the most critical customers promises we make. It means that customer data remains secure and private even if the machine storing that data is compromised.

An integral part of this promise is the ability to efficiently and securely manage the secrets used to access and encrypt this data. Our secrets management framework ensures that our secrets are acquired from credible sources, stored in secure locations, and are accessed only when authorized and through approved channels. It also alerts and detects on malicious usage or exfiltration of secrets. Much like we use JIT and JEA to govern access to service infrastructure and tasks against that infrastructure, JIT and JEA govern access to secrets stored in the service.

## Network Isolation

Just like our access control policies are designed to restrict privileged access to the service by our personnel, the goal of network isolation is to restrict the ability of different parts of the Microsoft 365 service infrastructure to communicate with each other except for the minimum necessary for the service to operate. Combined with our least privilege approach to access control, network isolation allows us to establish breach boundaries throughout the service.

Network isolation is not only about preventing unwanted authentication from one service partition to another. In addition to minimizing the damage that could be caused by compromised accounts, effective network segmentation is critical for defending against *unauthenticated* attacks such as the WannaCry outbreak of 2017. WannaCry was an enormously destructive attack globally, but it did not involve usage of privileged accounts or any kind of authentication: it exploited sensitive network paths that were left open between machines. Indeed, strong network isolation is a critical part of comprehensively protecting against lateral movement risks. The ability to establish a connection with a target, before authentication is even attempted, needs to be restricted as much as possible.

We closely scrutinize so-called RCE (remote code execution) ports – these are the ones attackers could exploit to pivot from one service to another and are what we mean when describing sensitive network path. Of course, defining RCE ports requires modeling of the service itself, as custom-built architecture may involve a unique set of RCE ports. We define network restrictions between partitions of the service that prevent all but required traffic from crossing partition boundaries. This requires us to deeply understand network traffic patterns between the services comprising Microsoft 365, and to restrict, at the host firewall and router layer, traffic that is not strictly necessary.

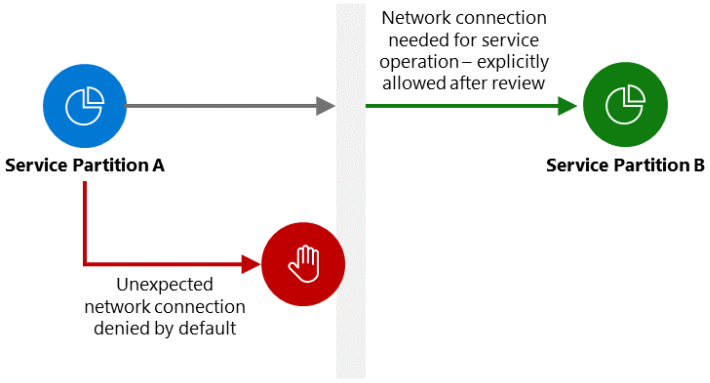


*Figure 2: network segmentation between service partitions. By default, network layer restrictions prohibit traffic between partitions. Only a restricted set of tightly controlled traffic is allowed following review*

We mentioned above that our access control systems are deeply integrated into the fabric of the service, ensuring that our access control mechanisms automatically scale with service growth, and our network isolation policies are no different. When core Microsoft 365 services grow, traffic from newly provisioned capacity into previously established parts of the service is denied by default. Teams must manually open network paths when necessary for a new service feature to operate, and we closely scrutinize any attempts to do so to ensure that only the minimal required paths are open. Again, our key security principles are at play here: even freshly provisioned capacity is not trusted to be secure, and our network isolation policies are automatically applied as the service scales.

## Security Monitoring and Response: Mitigating Risk if the Worst Happens

The assume breach model goes beyond designing architectural protections and access control policies: it means that no matter how effective those protections are, we cannot trust that they will always hold. We must assume a non-zero probability of successful attack, no matter how confident we are in our defenses.

We need to have the ability to detect and mitigate these attacks against the service infrastructure before they result in a compromise of customer data.

Our work in this space spans security monitoring and incident response:

- **Security Monitoring:** this is about building systems and processes to catch compromise to the infrastructure in real time and at scale, allowing us to respond to and stop attacks before they propagate throughout the service
- **Incident Response:** we need tools and processes to mitigate risk and evict attackers, also in real time and at scale, in response to the alerts raised by our monitoring systems

## Security Monitoring and Response at Scale

Automation, scale, and cloud-based solutions are key pillars of our monitoring and response strategy. For us to effectively catch and stop attacks at the scale of some of the Microsoft 365 core services, our monitoring systems need to automatically raise highly accurate alerts and to do so in real time – it is not enough to generate low-fidelity alerts and manually sort through large quantities of those. To that end, many of our alerts trigger immediate phone calls to a 24 x 7 on-call rotation (yes, even at 3am!). Similarly, when an issue is detected we need the ability to mitigate the risk at scale – we cannot rely on our team to manually fix issues machine-by-machine.

When possible, our systems automatically trigger an action to remediate the threats that have been detected. When that's not an option (for example, if the action needed to mitigate the attack is itself high-risk and therefore needs human oversight), our on-call engineers are equipped with a set of tools that enable them to act in real time and at scale to mitigate the threats that are detected.

We invest in two major areas to achieve these outcomes:

- **Robust, detailed security telemetry**: in addition to default server logging and application level data, core infrastructure in our service is equipped with customized security agents that generate detailed telemetry that we use for monitoring and forensics. We also consume telemetry from other Microsoft services that we depend on, resulting in a robust ecosystem of security signals for Microsoft 365 services.
- **Cloud-based monitoring and response tools**: telemetry from Microsoft 365 services is sent to a high-scale, real time system for centralized processing and alerting. When we need to take an action in response to an alert, or to further investigate forensic evidence throughout the service, our cloud-powered tools allow us to specify what needs to be done and where – and those actions take effect rapidly.

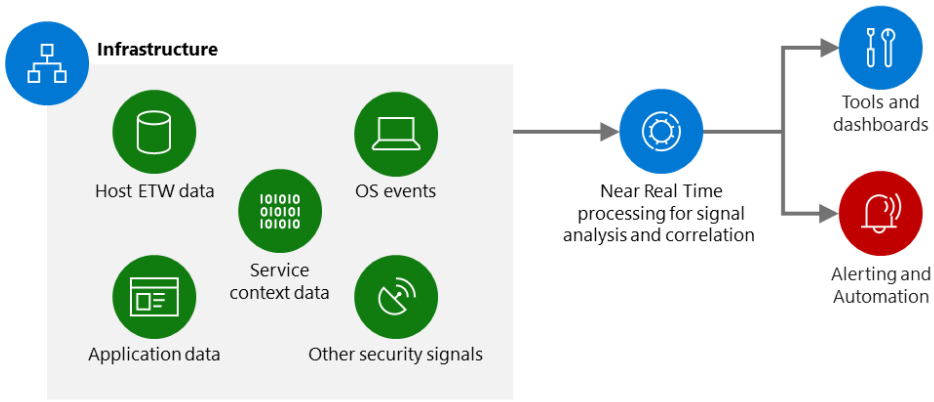


*Figure 3: Security Monitoring for service infrastructure. A robust set of signals is generated on the service infrastructure and is sent to a centralized real time pipeline for processing and analysis. The output is a set of high-fidelity alerts and signals that can be used for incident response and forensics*

Readers should note the emphasis on cloud-based, automated solutions for monitoring and response. We do not exclusively rely on local (that is, machine-specific) solutions. We also send telemetry from service infrastructure to a centralized pipeline that correlates the data and reasons over it to generate highly accurate alerts. For incident response, we do something similar: rather than only investigating issues on a per-machine basis, we rely on cloud-based tools which are integrated with the service fabric and inventory. This allows us to use those tools to iterate our incident response across machines in the service automatically. This is how we scale response, as localized approaches to security are simply not feasible for a fleet of one million machines. In fact, cloud-based approaches to telemetry and monitoring are an example of using scale to our advantage: scale results in an abundance of rich data at our disposal, which in turn enables us to build intelligent systems for monitoring and response.

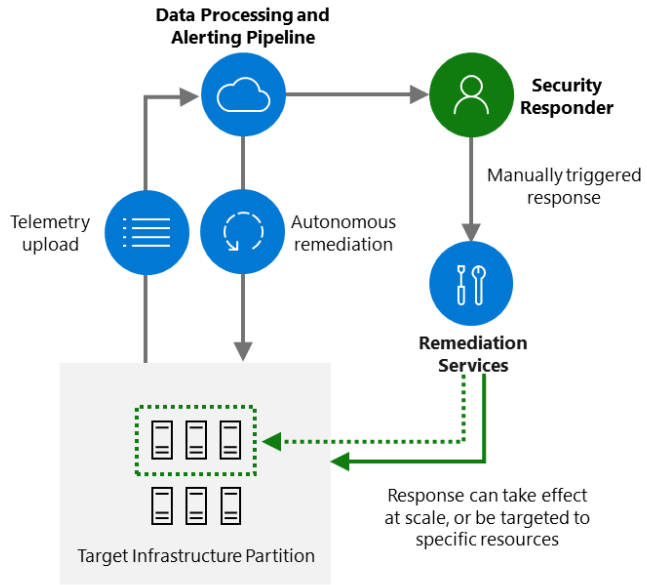

*Figure 4: Incident response is cloud-powered and service-aware. It can be triggered autonomously for basic actions, or manually for more complex scenarios. Remediation can take effect on a small number of machines, or across a service partition if necessary*

## Baseline Monitoring vs. Service Specific Scenarios

In the beginning of this document we mentioned the complexity of the service, and we wanted to describe how our approach to security monitoring addresses that. Effective security monitoring spans what we call "baseline scenarios", meaning generic attack patterns that do not exploit Microsoft 365 specific architecture or features, and scenarios involving exploitation of unique aspects of the Microsoft 365 services. It is not enough to monitor for baseline attacker patterns: we must assume that sufficiently motivated and sophisticated attackers will learn about service-specific architecture and will attempt to exploit that. This is certainly a major risk when it comes to insider threat scenarios, but we also see this as a risk in the context of nation state attackers and other APTs (Advanced Persistent Threats).

For baseline scenarios, we strongly rely on Microsoft's customer facing security products, such as Microsoft Defender ATP. For service specific attacks, we partner closely with engineering teams across the service to deeply model threats to the service architecture and build the telemetry and monitoring needed to mitigate those threats.

Earlier sections in this paper emphasized the fact that our security solutions are deeply integrated with our service – these products are aware of the nuances of service architecture and changes to that architecture. Security monitoring follows a similar model: our monitoring logic goes beyond baseline scenarios and incorporates deep awareness of service architecture and operations. Here are a few examples of service specific monitoring we build for Microsoft 365 service infrastructure:

- Modeling network traffic flows into and out of the service, raising alerts for deviations from normal patterns
- Analytical profiles for activities performed by internal operators who manage our service, alerting on deviations or suspected risky activity
- Abnormal use of privileged commands or functions that are only meant to be used for administration of the service

And there are many more such scenarios. The general point is that our services are one-of-a-kind: we've built a lot of unique systems and architecture to power the rich experiences we offer to our customers, but we assume that motivated attackers will also attempt attacks that exploit these unique capabilities. Our security monitoring must be prepared for that. Like all our service security solutions, security monitoring is integrated deeply with the service fabric itself.

## Constant Validation

Our assume breach principle is all about planning for the worst – given how seriously we take this philosophy, we would be remiss if we did not have a plan for mitigating potential gaps in our security posture. Indeed, we validate our security posture regularly, automatically, and through cloud-based tools (we hope that you notice a trend here).

We have two primary forms of validation:

- **Architectural and configuration assessment**: verifying that promises we make about our service architecture (for example, that specific networks are correctly segmented or that machines are up to date with required patches) hold and do not regress.
- **Post-exploitation validation:** simulating attacks directly against our infrastructure, with the goal of verifying that our monitoring and response systems work as expected in the production environment.

Both forms of validation result in a robust set of analytics and reports that allow us to quickly identify and remediate any failures. This is all generated automatically: the tools we use for validation are configured to run continuously and to target the entire service, or a representative sample when completeness is not feasible. Moreover, these are cloud based tools. Like our monitoring and response systems, our validation systems take a service-wide view. For example, our attack simulation tools are designed to intelligently navigate throughout the service infrastructure by using a cloud-based command and control system to coordinate actions.

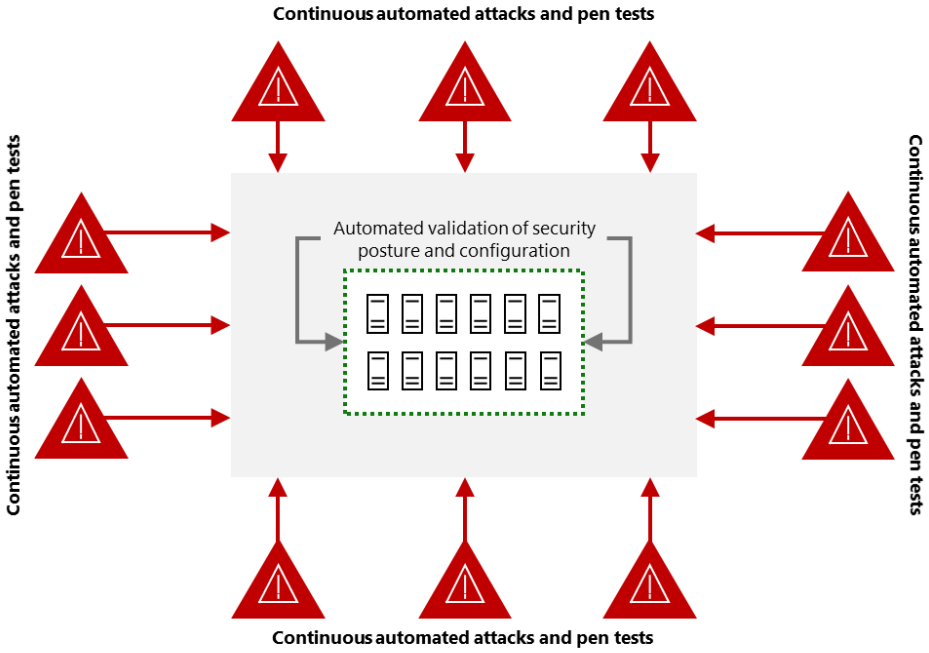


*Figure 5: Attacks are regularly simulated against the service, both through automation and through manual penetration testing. We also regularly scan the service architecture and configuration to ensure that our design principles are upheld.*

## Automated Assessment

No matter how well a system is designed, security posture can degrade over time: machines can go unpatched, unsafe changes to configuration can be introduced inadvertently, and regressions to security code can be introduced. We've built automation to continually assess our systems for this kind of degradation, so that we can immediately act to correct the issue. Our work here can be divided into two major categories:

- **Machine state scanning:** making sure that the machines comprising our infrastructure are up to date with the latest patches, and that their individual configurations are correctly set. This is often referred to as PAVC: patching, anti-virus, vulnerability, and configuration scanning.
- **Architectural validation**: As described earlier in this paper, a lot of our work focuses on designing service architecture to minimize the damage that an attack could cause. Network segmentation is a great example of this. But it's possible for network segmentation to degrade over time: if even a single machine in a service partition fails to receive an update (or a new highly powered service is introduced that listens on a previously unused port), an entire section of the service can be at risk of lateral movement through compromise of that initial machine. Another example is verifying that no accounts with standing administrative access have been introduced into the service. Our architectural validation work automatically identifies instances like this where the current state of the service has drifted for some reason from our desired state.

Broadly speaking, the goal of automated assessment is to ensure that the capabilities we described in the section on protecting service infrastructure function as expected. This is another application of our assume breach mentality. Implementing a protection is simply not enough. We need to regularly ensure that it works.

### Attack Simulation and Penetration Testing

In addition to automatically assessing the service for potentially dangerous misconfigurations, we put significant emphasis on simulating real attacks in our environment. We have an in-house penetration testing team that regularly conducts attacks against service infrastructure, and we also maintain an automated attack simulation system that regularly triggers small scale attacks. In other words, we attack ourselves constantly.

The goal of attack simulation is to validate our detection and response capabilities. Attack simulation often focuses on "post-exploitation" activities – while our penetration testing is certainly used to identify new vulnerabilities and paths into the service, a major priority is on what happens *after* the compromise. Are we detecting the attack quickly enough? Are we able to effectively remediate and evict the attackers? These are the key questions for our monitoring and response systems, and attack simulation allows us to answer those on an ongoing basis.

It's important to note that we attack our service constantly. We do not settle for infrequent penetration testing – attack simulation is a way of life for us. Indeed, as we've scrutinized recent high-profile breaches of other organizations, we've noticed that the impact of an attack is often exacerbated by insufficient validation of security controls (for example, lack of validation that monitoring systems work as expected can lead organizations to ignore or incorrectly respond to alerts that were raised during a real attack). Our goal with constant attack simulation is to make sure this does not happen to Microsoft 365. If the worst ever happens, we need to be prepared.

## Conclusion

Securing the infrastructure of one of the world's largest cloud services requires us to stay ahead of attackers while also keeping up with constantly increasing service scale and complexity. Maintaining customer trust in Microsoft 365 requires us to design our services to a robust set of core security principles and to make sure those principles are embedded deeply into service design and operations. This paper

has provided a look into what this means, and we'll expand on this and other security topics critical to our business in future papers. We hope that this paper, and the others we publish in this series, will give our customers additional insight into just how seriously we take our mission to secure their data.