opentext™ | Cybersecurity

Position Paper

Getting the Most out of NetIQ Access Manager

With NetIQ Access Manager, administrators can control the user experience to a level that few other technologies can match. This position paper reviews ways the access gateway can be used in complex environments as well as the best practices for applying its capabilities.

Table of Contents

Access Management Best Practices

As the demands of secure web access become more complex, organizations are finding that they often need more than just federation, especially as consumption of a wider set of services and resources continues to be more diverse. This position paper reviews various ways that NetIQ Access Manager by OpenText™ can be used to deliver the best access experience with the right level of security required.

This position paper reviews various ways that NetIQ Access Manager by OpenText™ can be used to deliver the best access experience with the right level of security required.

Solve Problems with Access Manager's Reverse Proxy

As organizations design their environment, questions about the use of the Access Gateway reverse proxy are common when planning an NetIQ Access Manager implementation. Some of the questions we get most often are:

- Gs a reverse proxy a choke point? How can it handle the load?
- I already have a reverse proxy. Why do I need another one?
- Should applications that integrate with the Identity Server using federation protocols be placed behind the Access Gateway?
- Why doesn't NetIQ Access Manager provide agents for integration?
- Should caching be enabled on the Access Gateway reverse proxy?
- Should the Identity Server be placed behind the Access Gateway?
- Should I avoid the Access Gateway to minimize operational issues with the application support teams?
- Are the benefits of using the Access Gateway worth the extra infrastructure and complexity when Identity Injection and Form-Fill are not needed?

This whitepaper will attempt to answer these questions and provide guidance on when and how to utilize the Access Gateway. Before we get to these questions, we need to address two more fundamental questions:

- What is the Access Gateway?
- What are the functions and benefits of the Access Gateway?

NetIQ Access Manager Architecture— What Is the Access Gateway?

NetIQ Access Manager has four high-level functional components: the Administration Console, the Analytics Server, the Identity Server, and the Access Gateway.

The Four Components of NetIQ Access Manager

- Administration Console
- Analytics Server
- Identity Server
- Access Gateway

The **Administration Console** component is used to manage, store, and distribute the configuration of the system. Multiple Administration Consoles share a replicated data store, making the system inherently fault tolerant and eliminating single points of failure.

The **Analytics Server** captures audit and operational data. It provides a configurable dashboard and powerful reporting capabilities. It provides visibility into the security, utilization, and performance of your access management system.

The **Identity Server** is the service responsible for authenticating users to access web workloads integrated with NetIQ Access Manager. It is also commonly referred to as an Identity Provider (IDP) because it authenticates users using federation protocols. It supports all common federation protocols and also has the ability to broker federation connections. The authentication framework utilizes a plug-in architecture, so adding custom authentication methods is easy if one of the out-of-the-box options doesn't meet your needs. While the Identity Server requires an Administration Console, it does not require the use of the Access Gateway.

The fourth component is the **Access Gateway** (AG). It can be described as a federation-enabled reverse proxy that relies on the Identity Server for user authentication to provide single sign-on, authorization control, and remote access. That description says a lot without saying much all at the same time. While the function and purpose of the Administration Console and the Identity Server are easy to grasp, there is often a lack of understanding about the functions and purpose of the Access Gateway. The next section will explain those functions and illustrate the benefits that the Access Gateway provides.

Before we go further, it's important to understand the technical architectures used by Web Access Management (WAM) systems. In the early days of WAM, there were two distinct approaches to integrating with and securing applications.

Agents

The first approach was the use of "agents." These were small pieces of software that would be added to the application or the application platform. The agent would communicate with the access management system and enforce authentication and, in some cases, authorization. These agents were proprietary to each WAM system because there were no standardized protocols for application-to-WAM system communication.

It's important to understand the technical architectures used by Web Access Management (WAM) systems. In the early days of WAM, there were two distinct approaches to integrating with and securing applications:

- Agents
- Reverse Proxy

Pros:

- No additional component in the request flow
- Avoids the need for URL rewriting
- Requests are direct from browser to application, no additional components
-

Cons:

- Agent proliferation, upgrades can be a nightmare
- Tight integration with the application, often requiring application modifications
- Requires "foreign" software on the application platform
- Any agent-enforced authorization is performed on the application platform
- Agents require a centralized control point in the form of a policy server, which are difficult to scale and manage
- Single Sign-On (SSO) is often limited to single cookie domains, eliminating cross-domain SSO

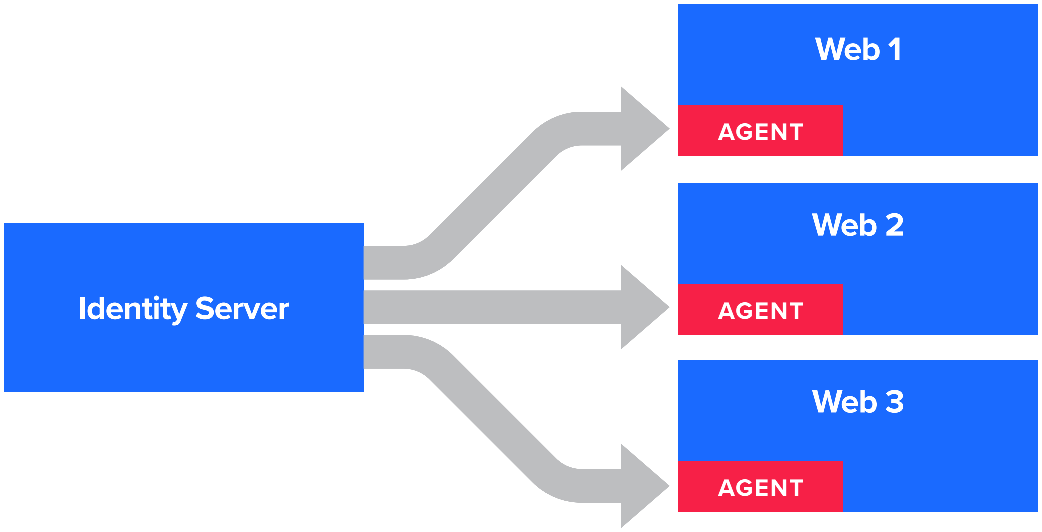


Figure 1.

Reverse Proxy

The second approach to integrating with and securing applications was to use a reverse proxy in front of the application. The proxy would communicate with the other components of the WAM system. You could think of it as a "centralized" agent that runs on a separate server or cluster of servers. It would provide authentication information to the application by either adding HTTP headers to the request or by automatically filling and submitting the application's standard login form.

Pros:

- Does not require application modification
- Very fast to implement
- Caching can be used to improve performance
- No foreign software on the application platform
- Authorization can be enforced at the proxy
- Centralized logging
- Application server is not directly accessible from the public network

Because the Access Gateway is a natively federation-enabled reverse proxy, the applications that it protects benefit from the Identity Server's federation support, essentially federation enabling protected applications.

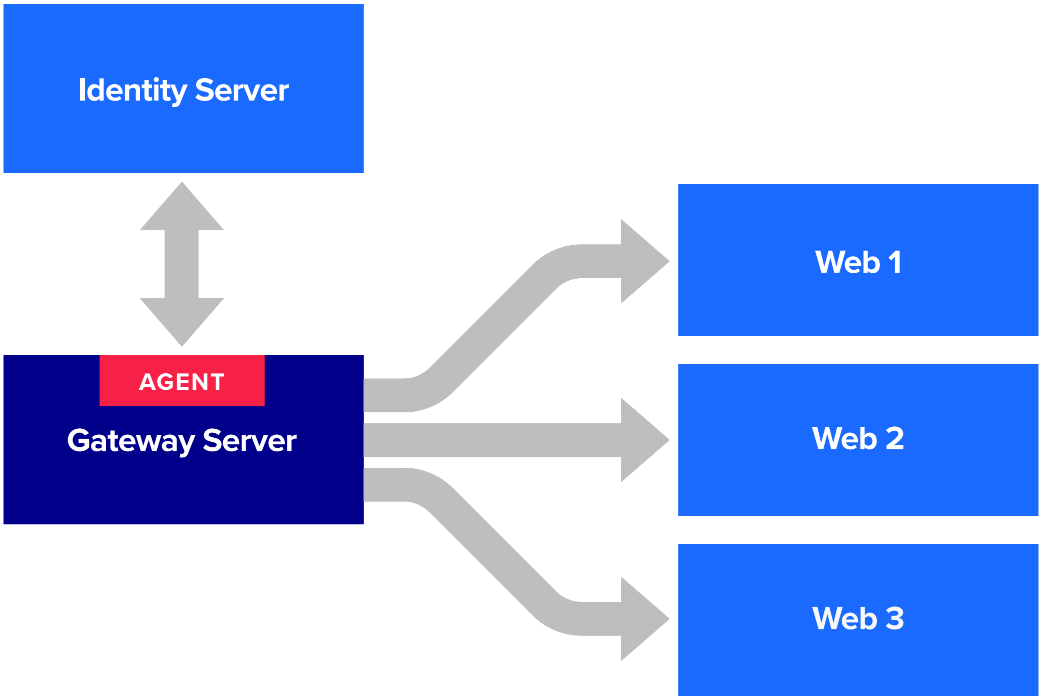


Figure 2.

The Access Gateway is a reverse proxy whose most basic function is to provide access to legacy applications that don't have the ability to communicate directly with the Identity Server using federation protocols. Because the Access Gateway is a natively federation-enabled reverse proxy, the applications that it protects benefit from the Identity Server's federation support, essentially federation enabling protected applications. NetIQ Access Manager was originally implemented almost exclusively using the proxy-based approach since the adoption of federation was in its infancy and initial adoption was slow and not well understood, especially in typical COTS applications. In fact, NetIQ Access Manager evolved from an incredibly scalable internet caching proxy that was marketed with the goal of reducing bandwidth requirements, because high-speed connections were either very expensive or entirely unavailable at the time. Of course, the competition went to great lengths to convince people that their agent-based approach was superior. The reality is that each approach provides distinct benefits and is why many solutions that were previously agent-based have also embraced reverse proxy approaches.

Unlike when NetIQ Access Manager was first introduced, support for federation protocols (such as SAML and OAuth2 / OpenID Connect) is now much more common in COTS applications. It is almost a given for modern cloud-based SaaS and mobile applications, eliminating the need for proprietary agents. NetIQ Access Manager does not provide proprietary agents, simply because they are no longer needed. The standard protocols are a much better option.

If the application, or application platform, is able to use protocols such a SAML2, OAuth, or WS-Federation, then the Access Gateway is optional. However, there are a number of reasons why you want to include it.

What Are the Functions and Benefits of the Access Gateway?

These evolutions have led many to ask why there is a continued need for a reverse proxy since proprietary agents are no longer required. As described earlier, there are benefits to using a proxy-less approach, but the Access Gateway does provide functionality beyond just enabling integration with the application. The obvious use for the proxy is in a situation where you cannot reconfigure the application or add an agent, for whatever reason. Beyond integration to legacy applications, the Access Gateway can be used to:

- Enhance the capabilities of applications that do integrate directly with the Identity Server
- Provide an additional layer of security with advanced capabilities
  - Minimize vulnerabilities by shielding application platforms
  - Enforce centralized, policy-based access control that can replace or enhance that provided by your application
  - Provide secure remote access to legacy web workloads that still exist behind the corporate firewall
  - Provide up-to-date TLS capabilities
- Cache web content, enhancing system performance
- Route requests based on the requested path
- Centralize logging, monitoring, and analytics
- Provide integration from the Identity Server to legacy applications that don’t have the ability to integrate directly

Enhance the Capabilities of Applications that Integrate Directly with the Identity Server
If the application, or application platform, is able to use protocols such a SAML2, OAuth, or WS-Federation, then the Access Gateway is optional. However, there are a number of reasons why you want to include it. The Access Gateway provides an additional layer of security. It likely provides a smaller attack surface than your application servers, reducing possible vulnerabilities. Patching and vulnerability mitigation is simplified because you don’t

need to consider the dependencies of your application. The reverse proxy can cache web content, enhancing system performance. It can also route traffic based on the requested path, so you can deliver multiple back-end applications and services as a single unified application. Finally, the Access Gateway provides centralized, policy-based access control that can replace or enhance that provided by your application.

Provide an Additional Layer of Security with a Minimized Attack Surface

Most application delivery infrastructures are a mix of different platforms and versions. Ensuring that all vulnerabilities are patched and remediated across all your platforms and versions can be problematic. In some cases, you might not be able to patch or upgrade because of compatibility or support issues. By routing requests through a reverse proxy, such as the Access Gateway, you can reduce exposed vulnerabilities and safely operate otherwise risky platforms. It can also provide secure, up-to-date, TLS communication for applications and systems that can't be upgraded or are in the process of being replaced. The reverse proxy enforces TLS 1.2 communication between clients and the Access Gateway on uncontrolled public networks. But it can also support legacy SSL communication to origin web servers where the communication channel can be secured by internal network security controls and firewalls. Having a single, easily updated platform exposed to the public network makes your entire system more secure.

By routing requests through a reverse proxy, such as the Access Gateway, you can reduce exposed vulnerabilities and safely operate otherwise risky platforms.

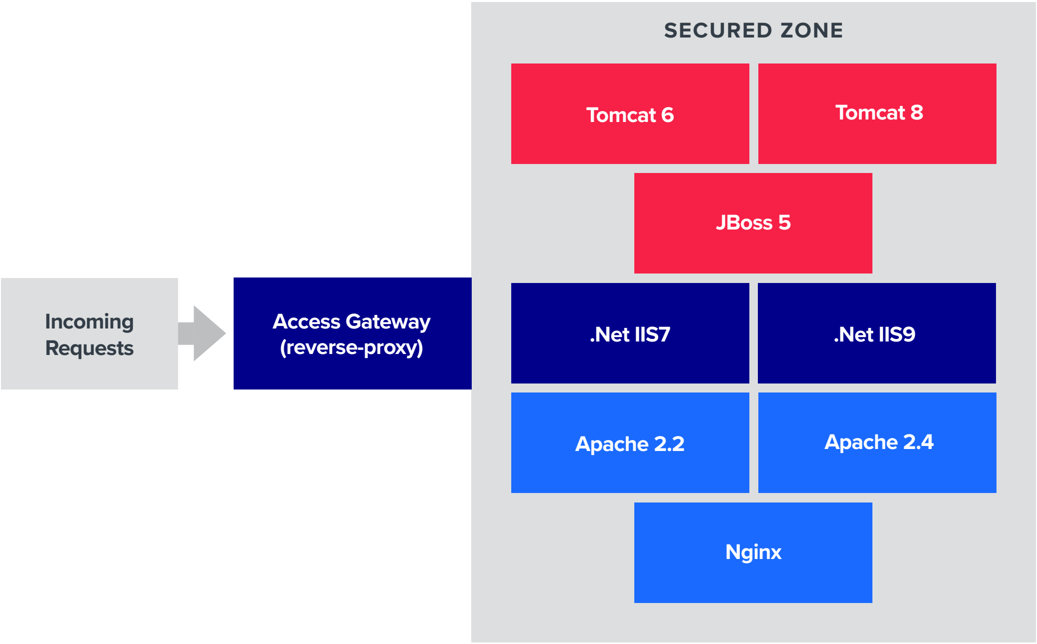


Figure 3.

In addition to the extra security, the Access Gateway can route requests based on the URL path requested. This means that you can serve content from any of your back-end platforms using a common domain name and even context, enabling you to deliver multiple back-end applications and services as a single, unified application.

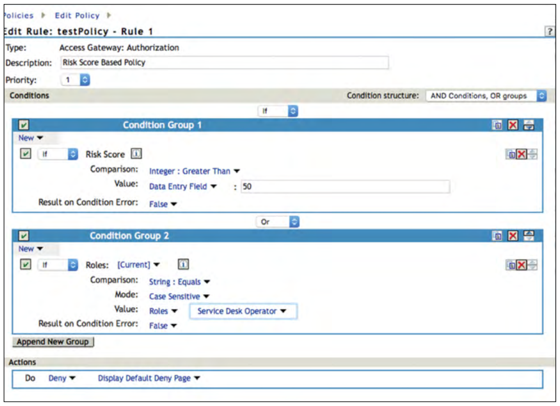
The Access Gateway can also apply authentication and authorization policies on a per-path basis. While the application platform might have a similar capability, it is often either hard to manage or incomplete. For example, the widely-used WordPress content management system exposes APIs on specific paths, yet it does not provide effective ways to secure them. These APIs have been repeatedly targeted for attack. By establishing policies on the Access Gateway, which controls access to the specific paths associated with the APIs, you can eliminate an entire class of vulnerabilities.

Where the platforms do provide effective authorization mechanisms, there is still the issue that each platform has its own method for administering authorization. Management of access gets even more complicated because, for most platforms, the configuration is specific to each server and must be administered, and audited, separately. By enforcing access policies centrally, the Access Gateway simplifies administration and makes auditing easy.

Another example of how the Access Gateway can enhance the security of applications involves step-up authentication or simply selecting an authentication method. Most application platforms provide very basic abilities to configure authentication. Typically, they support either authenticated or public access where authenticated access is done with a single authentication method. They don't have an easy way to specify that access to a specific resource requires stronger authentication or that different authentication methods should be used for different categories of users. Access Gateway policies can do all of this easily.

Better control of authentication is just the start. The Access Gateway also provides powerful authorization policies, including risk-based authorization. These polices are easy to manage and allow a level of control that the native abilities of the application platforms can't match.

The Access Gateway also provides powerful authorization policies, including risk-based authorization. These polices are easy to manage and allow a level of control that the native abilities of the application platforms can't match.



olicies ▸ Edit Policy ▸

dit Rule: testPolicy - Rule 1

Type: Access Gateway: Authorization

Description: Risk Score Based Policy

Priority: 1

Conditions — Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

New

☑ If Risk Score

Comparison: Integer : Greater Than

Value: Data Entry Field : 50

Result on Condition Error: False

Or

☑ Condition Group 2

New

☑ If Roles: [Current]

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles Service Desk Operator

Result on Condition Error: False

Append New Group

Actions

Do Deny Display Default Deny Page

Figure 4.

Cache Web Content, Enhancing System Performance

The Access Gateway can be used to cache static content. Doing so can greatly increase performance and reduce application server resource requirements. So, if caching is so great, why do some administrators prefer to globally disable it? Applications are responsible for flagging content that should not be cached, and the proxy does honor the application's request by default. However, some applications do not flag their content appropriately. There are others that do some strange things that stretch the HTTP specification. Rather than give up the benefits of caching completely, it's better to leave it enabled and identify the specific resources that should not be cached, and then configure exceptions for them.

Route Requests Based on the Requested Path

Path-based routing of requests enables you to deliver multiple back-end applications and services as a single domain and even context. This allows for a complex, multiplatform, web infrastructure to be presented as a single composite application.

Centralized Logging, Monitoring, and Analytics

The Access Gateway proxy can also be used to provide centralized request logging that can either supplement or replace any logging done on the application platform. Capturing request logs at the proxy is often simpler than trying to consolidate logs from the multiple application platforms found in most environments. The proxy logs can also capture information about the request that might not be available in the application logs.

The Access Gateway proxy can also be used to provide centralized request logging that can either supplement or replace any logging done on the application platform.

NetIQ Access Manager includes an analytics server that can be used to collect and visualize data from the Access Gateways and from the Identity Servers.

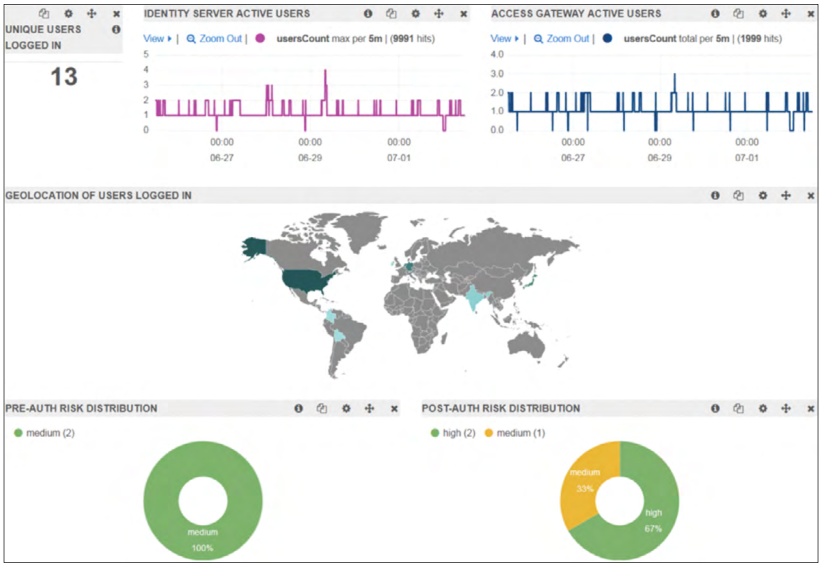

UNIQUE USERS LOGGED IN
13
IDENTITY SERVER ACTIVE USERS
View ▸ | Zoom Out | ● usersCount max per 5m | (9991 hits)
5
4
3
2
1
0
00:00 06-27
00:00 06-29
00:00 07-01
ACCESS GATEWAY ACTIVE USERS
View ▸ | Zoom Out | ● usersCount total per 5m | (1999 hits)
4.0
3.0
2.0
1.0
0.0
00:00 06-27
00:00 06-29
00:00 07-01
GEOLOCATION OF USERS LOGGED IN
PRE-AUTH RISK DISTRIBUTION
● medium (2)
medium 100%
POST-AUTH RISK DISTRIBUTION
● high (2) ● medium (1)
medium 33%
high 67%

Figure 5.

The benefits described above emphasize why it may make sense to implement the Access Gateway, even if it's not needed for integration between the application and the Identity Server. Enhanced security, centralized administration, centralized logging, composite applications, and increased performance are all desirable benefits. You need to carefully consider whether the simplicity of non-proxied configuration outweighs these benefits for each application in your environment.

Provide Integration from the Identity Server to Legacy Applications That Don't Have the Ability to Integrate Directly

In some cases, there is no practical way to use modern Access Management protocols with an application. This is particularly common with legacy applications or with commercial applications that explicitly refuse to support such integration. The Access Gateway provides a solution to this problem because it can be used to automatically fill and submit the applications login from or it can inject headers with user-identifying information.

With all those benefits, why would anyone choose to implement the Identity Server without using the Access Gateway proxy?

Should the Identity Server Be Placed behind the Access Gateway?

There is another set of questions that often come up regarding the placement of the Identity Server behind the Access Gateway proxy. If you choose to implement the Access Manager Appliance, then the choice is made for you because the Identity Server is accessed through the proxy that is running on the same appliance node.

There are real benefits from protecting the Identity Server with the Access Gateway:

- A common domain name can be used, which also means a common certificate without the need to use a wildcard certificate.
- A single load balancer configuration can be used.
- The proxy can cache static content, which can improve Identity Server performance.
- The logging provided by the Apache proxy process is more extensive than that provided by the Identity Server's Tomcat container.
- Connection setup is between the proxy and the browser, removing load from the Identity Server. The connection process that the proxy uses to the Identity Server is optimized for efficiency and connection reuse.
- Using the proxy means the Identity Server is not directly exposed to the public network, providing all the security benefits detailed earlier.
- You're using a configuration similar to the Access Manager Appliance.

With all those benefits, why would anyone choose to implement the Identity Server without using the Access Gateway proxy? The most common reasons are:

- Use of an authentication method, such as X.509 certificate authentication, which requires SSL/TLS renegotiation.
- Concern about increasing the resource requirements of the Access Gateway.
- Concern about adding complexity and another point of failure because the Identity Server is now dependent on the Access Gateway.

The first point is a valid concern since such authentication methods typically require a direct connection from the browser to the Identity Server. There are ways to configure the proxy to enable these methods to work but doing so adds complexity. However, these authentication methods are rarely used and are relatively complex to manage in any case. If such methods are needed, the preferred approach is to utilize NetIQ Advanced Authentication by OpenText™.

Prior to NetIQ Access Manager version 3.2, when the Access Gateway was a 32-bit system, the resource requirements of the Access Gateway were a valid concern. There were customers running large-scale implementations that were hitting the limits of a 32-bit architecture. Today, NetIQ Access Manager is a fully 64-bit architecture and is so massively scalable that the addition of the Identity Server traffic is not a problem. While it's true that having the proxy in the request flow adds dependency and complexity, most implementations already have this dependency, since they have applications behind the proxy. Even if they don't, the benefits provided by the proxy outweigh the drawbacks for most implementations. We now consider placing the Identity Server behind the Access Gateway the preferred configuration.

There is still a need for the different groups to work together effectively when an issue occurs. Avoiding the Access Gateway is not a silver bullet for organizational skillset issues.

Should I Avoid the Access Gateway to Minimize Operational Issues with the Application Support Teams?

In most organizations, the WAM system and the applications are owned and supported by distinct, somewhat antagonistic, groups. This can lead to tension and conflict when issues arise since it can be difficult to isolate the root cause of the issue. Often, the application support team does not have the knowledge and experience required to deeply understand the WAM and network infrastructure. Likewise, the WAM support group often does not have the knowledge and skills to understand the application and its infrastructure. This tension has led to both groups wanting to make the line between WAM and application as clear as possible. Removing the proxy from the request flow is seen as a way to do so. It's seen as an action that can be taken without having to tackle the difficult task of solving organizational and skills issues. While removing the proxy might eliminate a subset of possible issues, there is still integration between the Identity Server and the application. There is still a need for the different groups to work together effectively when an issue occurs. Avoiding the Access Gateway is not a silver bullet for organizational skillset issues.

I Already Have a Reverse Proxy. Why Do I Need Another One? Can I Use the Access Gateway in Place of My Current Proxy?

The benefits provided by a proxy have led to proxies being used as a standard part of application delivery architectures. The downside of this trend is that it's not uncommon to see two or even three layers of proxies between the browser and the application server. This results in complexity, inefficiency, and higher operational costs.

Sometimes the application proxies are proprietary and application-specific, so they can't be easily removed from the architecture. But in most cases, they are standard HTTP proxies that are pooling connections and caching static content. Since the Access Gateway provides these same services, you might be able to remove the secondary proxy from the request flow. We recommend that you analyze any proxies used for application delivery to determine if they could be replaced by the Access Gateway proxy.

The Access Gateway is a better choice than other proxies because it provides centralized management and clustering features that most stand-alone proxies do not. Access Gateway provides enforcement of powerful authorization and authentication policies based on dynamic factors—a capability that is not found on typical proxies. The monitoring and instrumentation features of the Access Gateway, especially the Analytics Server, enhance your ability to support and secure your system.

The Access Gateway is a better choice than other proxies because it provides centralized management and clustering features that most stand-alone proxies do not.

Is A Reverse Proxy a Choke Point? How Can It Handle the Load?

A reverse proxy that is not properly scaled for the load can negatively impact performance. However, when configured properly, a caching proxy can improve performance while reducing resource requirements on the application servers. Scaling the proxy is usually easier and less expensive than scaling the application servers. Most current, high-performance application delivery architectures utilize caching reverse proxies.

The current 64-bit architecture of the Access Gateway is incredibly scalable, both vertically and horizontally. Testing and real-world implementations have shown that the system can support massive loads. See the test results B2C implementation guide and the scaling

Why Doesn't NetIQ Access Manager Provide Agents for Integration?

Unlike when NetIQ Access Manager was introduced, support for federation protocols (such as SAML and OAuth2 / OpenID Connect) is much more common in off-the-shelf applications and almost a given for modern cloud-based SaaS and mobile applications, eliminating the need for proprietary agents. NetIQ Access Manager does not provide proprietary agents simply because they are no longer needed. The standard protocols are a much better option.

Should Caching Be Enabled on the Access Gateway Reverse Proxy?

The Access Gateway can be used to cache static content. Doing so can greatly increase overall performance and reduce application server resource requirements. So, if caching is so great, why do some administrators globally disable it? Applications are responsible for flagging content that should not be cached, and the proxy does honor the applications request by default. However, some applications do not flag their content appropriately. There are others that do some strange things that stretch the HTTP specification. Encountering such issues has led some to turn off caching as a quick fix. Rather than give up the benefits of caching completely, it's better to leave it enabled and identify the specific resources that should not be cached, and then configure exceptions for those resources.

Are the Benefits of Using the Access Gateway Worth the Extra Infrastructure and Complexity When Identity Injection and Form-Fill Are Not Needed? Should Applications That Integrate with the Identity Server Using Federation Protocols Be Placed Behind the Access Gateway?

These two questions share a common answer. This paper has discussed the many benefits realized through the use of the Access Gateway. Security, performance, and functionality can improve when you have a high-performance, identity-aware, reverse proxy between the browser and the application. The requirements and operational considerations of each application should be evaluated to determine the best option for your implementation.

As a best practice, we recommend that the Access Gateway be used for internally-controlled and supported applications. The exception would be when the application operations team is so organizationally distant from the NetIQ Access Manager operations team that it makes sense to treat the application as if it were an external application.

> This paper has discussed the many benefits realized through the use of the Access Gateway. Security, performance, and functionality can improve when you have a high-performance, identity-aware, reverse proxy between the browser and the application.

Learn More

To find out more about NetIQ Access Manager visit our product web page at:
www.microfocus.com/en-us/cyberres/identity-access-management/access-manager

About NetIQ

NetIQ provides security solutions that help organizations with workforce and consumer identity and access management at enterprise-scale. By providing secure access, effective governance, scalable automation, and actionable insight, NetIQ customers can achieve greater confidence in their IT security posture across cloud, mobile, and data platforms.

Visit the NetIQ homepage at **www.cyberres.com/netiq** to learn more. Watch video demos on our NetIQ Unplugged YouTube channel at **www.youtube.com/c/NetIQUnplugged**.

NetIQ is part of Cybersecurity, an OpenText™ line of business.

Connect with Us

www.opentext.com

opentext™ | Cybersecurity

OpenText Cybersecurity provides comprehensive security solutions for companies and partners of all sizes. From prevention, detection and response to recovery, investigation and compliance, our unified end-to-end platform helps customers build cyber resilience via a holistic security portfolio. Powered by actionable insights from our real-time and contextual threat intelligence, OpenText Cybersecurity customers benefit from high efficacy products, a compliant experience and simplified security to help manage business risk.

762-000048-003 | O | 11/23 |