Inside Citrix chapter twenty one – The one with the user login process

Now that we have looked at the various individual FMA components and services involved, let's have a closer look to see how everything works together. What happens when a user logs in, authenticates and so on.

Resource enumeration

When a user logs in for the first time, meaning that there are no active and/or disconnected sessions lingering around somewhere, right after the user is authenticated the resource enumeration process kicks in and will eventually show the user its assigned resources. That's why the user authentication process and resource enumeration basically go hand in hand. Let's put the two together and see what happens, I'll only use StoreFront in my examples.

Resource subscription

Due note that when using StoreFront users will first have to select and subscribe to resources (applications and/or desktops) from the store before they show up on their main home screen and can be launched. When using so called Keywords, Administrators can pre-subscribe users to certain core resources so that their home screen won't be completely empty when logging in for the first time. This also works for assigned Desktops. Use Keywords: auto with the application and/or desktop of choice.

The user login process

When troubleshooting a XenDesktop environment/architecture it's important to know which components and services are involved, how they interact and what is supposed to happen during normal operations. In fact, I guess it's safe to state that that goes for all problems in life: if you don't know or understand the basics of what you are dealing with, then you're bound to get lost, fast.

Throughout the next few sections I will zoom in on the user login and resource enumeration process as well as the steps needed to actually launch a desktop and/or application and talk about what happens during some of the most common day-to-day operations and processes, so common that in most cases we don't even think about what's going on under the hood until… it stops working!

I have already discussed the user authentication process from a StoreFront and Web Interface perspective, but just to be sure we are all on the same page I'll highlight them here as well.

User authentication

With StoreFront, users are authenticated by the authentication service, which is an integral part of StoreFront. Users can authenticate to StoreFront using different methods: usernames and passwords, domain pass-through, NetScaler pass-through, using smart cards or by enabling unauthenticated user access. As soon as a user logs in by filling in his or her username and password (on the StoreFront web page using the so-called Receiver for website configuration, or using a locally installed Citrix Receiver), the StoreFront authentication service will pick up the user credentials and authenticate them with a domain controller.

Once authenticated, StoreFront will forward the user credentials, as part of an XML query, to one of the configured Delivery Controllers, assuming you configured at least two, of course. In between, StoreFront will check its local datastore for any existing user subscriptions and store them in memory. Next, the Delivery Controller receiving the credentials will again contact a domain controller, this time to validate the user's credentials before responding to the StoreFront server.

If we look at Web Interface, user authentication works a bit different, since it has no internal authentication service. When a user logs in by filling in his or her username and password, Web Interface will immediately forward these credentials, as part of a XML query, to a Delivery Controller where a domain controller will be contacted to authenticate the user before responding to the Web Interface server.

As before, Web Interface can also authenticate users to, and enumerate and aggregate resources from, multiple XenApp Farms and XenDesktop Sites; no App Controller, though.

> **FMA fact**: Knowing the architecture, the components, the way traffic flows throughout and expected behaviour is the only way to successfully troubleshoot your FMA-based infrastructure.

StoreFront 3.x

As of StoreFront 3.0 Citrix reintroduced XML-based authentication. By simply running a few PowerShell scripts, user authentication falls back to the XenDesktop / XenApp XML service, as with Web Interface, useful when StoreFront is not in the same Domain as XenDesktop / XenApp or when it is not possible to set up an Active Directory trust, for example. Again, this method will be disabled by default, at least now you have options.

External user authentication through NetScaler

Let's take it step by step and see what happens when someone logs in externally. I'll assume that your NetScaler Gateway is set up and configured to integrate your StoreFront server(s), you have Receiver installed, SSL certificates are present, and that a STA / XML / Broker service address (Delivery Controller) and a domain controller for authentication purposes are also configured.

Perhaps you want to load balance your StoreFront and/or Delivery Controllers by creating and configuring virtual load balance servers on the NetScaler, adjust the theme of the NetScaler's Web Interface and finally, you'll probably have to configure your StoreFront deployment to accept pass-through authentication from NetScaler. Where port Nr. 443 (SSL) is used you can also use port 80, although 443 is recommended.

1. A user opens a web browser and connects to the external URL of the NetScaler Gateway (preferably using SSL over port Nr. 443). Here he or she will fill in his or her username and password. A locally installed Citrix Receiver can also be used to establish a direct connection to the NetScaler Gateway. Citrix Receiver uses so called Beacons to determine if a connection is internal or external and handles it accordingly. Check the (red) link for some more detailed information around Beacons and the discovery process.
2. During the login/authentication process an EPA (End Point Analyses) scan might be performed as part of a SmartAccess/SmartControl policy, for example, or NetScaler multi-Factor a.k.a. nFactor authentication could be configured (optional as of NetScaler 11.0 build 62.x and onwards).
3. Eventually the NetScaler will authenticate the user credentials (session ticket) against Active Directory, preferably using TCP port Nr. 636 (SSL) based upon the configured Authentication Policy. This could also involve two-factor/RADIUS authentication, which is basically considered a must have/minimum these days. Like StoreFront, the NetScaler has its own Authentication Service.
4. Once authenticated, the NetScaler will assign a session cookie (note that it does not built/assign the authentication token as part of the initial authentication process), which will be used for any potential subsequent client requests.
5. Next the user session and the user authentication credentials get redirected to StoreFront (based upon the configured Session Policy) where it will perform a call-back to the NetScaler (Gateway Virtual Server) that handled authentication to validate the user in the first place. The authentication details will then be send to the StoreFront Authentication Service, which is similar to the Authentication Service of the NetScaler mentioned earlier.
6. This is where the earlier mentioned authentication token is built/generated — by default the StoreFront Authentication Service will take care of this. However, as of StoreFront version 3.0, Citrix re-introduced XML-based user authentication. By simply running a few PowerShell scripts user authentication falls back to the XenDesktop/XenApp XML service, which is equal to how Web Interface used to handle things. Particularly useful when StoreFront is not in the same domain as XenDesktop / XenApp and when it is not possible to set up an Active Directory trust, or multiple. Just be aware that this method will be disabled by default. As of StoreFront version 3.5 and upwards PowerShell is no longer needed to enable XML based user authentication, it can be enabled and disabled directly from the StoreFront management console.
7. From here the user credentials will be forwarded, as part of a XML query, to the configured Broker (XML) service on one of the available Delivery Controllers. Both these transactions will use port Nr. 80 by default, which of course can be changed to 443 (SSL).
8. In between, StoreFront will check its local data store for any existing recourse subscriptions and stores these in memory.
9. The Broker (XML) service will again contact a domain controller (using port Nr. 389 by default, change to 636 for SSL) to validate the user credentials, note that this is different to the user authentication process, as we've established earlier. During this process, it will find out to which security groups (SID's) the user belongs.
10. You basically authenticate/validate against LDAP three times:

11. Through NetScaler (session cookie) -> Active Directory, followed by a redirection of the authentication credentials over to the StoreFront server.
12. Through Storefront, either using the SF Authentication Service or via SF to the XML Service on one of your Delivery Controllers -> Active Directory, this will generate/built the authentication token.
13. Through the XML Service (validation) -> Active Directory, to find out the accompanying security group SID's used for resource enumeration.
14. With this information the Delivery Controller, or Broker (XML) service will contact the Central Site Database to find out which resources have been assigned to the user. It does this over port Nr. 1433 / 1434.
15. This data will then be gathered and send back to the StoreFront server in the form of an XML formatted file, through/using the Broker (XML) service.
16. Based on this information StoreFront will generate a web page containing all the assigned resources, which will be routed through the NetScaler Gateway and presented to the user. The users home screen will be populated with any pre-subscribed resources (Keywords). Depending on how you connected, your resources will be displayed either directly using a Receiver for web sites or you'll find them within the locally installed Citrix Receiver instead. The user will be able to browse its own personal app store for any assigned resources to which he or she can subscribe and then launch.

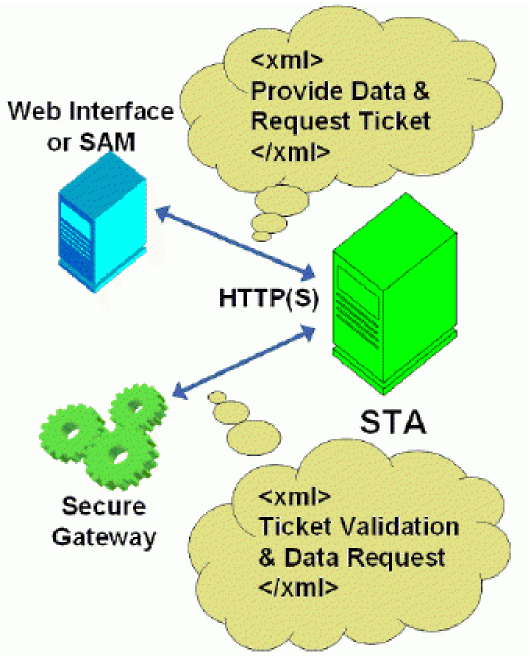
> **FMA fact:** If you don't enable authentication on the NetScaler's login page the NetScaler will contact StoreFront and the user will be presented (through the NetScaler) with the StoreFront login page (Receiver for web sites). The user fills in his or her credentials and authentication will be handled by StoreFront.

The (external) launch process

Here we basically pick it up where we left off at the end of the resource enumeration process as explained above. Just as with the authentication process, there are some differences in how a recourse is launched with, and without a NetScaler in between. Also, when launching a Hosted Shared Desktop (XenApp) or a published application, as opposed to a VDI virtual machine (XenDesktop) there is an extra load balance step involved as well. Let's see what happens when we launch a published Hosted Shared Desktop trough NetScaler.

The Secure Ticket Authority

Before we continue… You might have heard about something called the STA, or the Secure Ticket Authority in full. It was first introduced with one of the earlier Secure Gateway editions over twelve years ago. It (the STA) runs as a service and is part of the Broker Service on the Delivery Controller just like the XML service. During the resource launch sequence the StoreFront server as well as the NetScaler will both need to be able to communicate with the STA. As such you need to configure the NetScaler and the StoreFront server(s) or Web-Interface server(s) to point to the exact same XML/STA service(s)/Deliver Controller(s).


Web Interface or SAM
<xml> Provide Data & Request Ticket </xml>
HTTP(S)
STA
Secure Gateway
<xml> Ticket Validation & Data Request </xml>

> **FMA fact:** The NetScaler Gateway uses the STA to guarantee that each user is successfully authenticated. If users have valid STA tickets, the gateway assumes that they passed the authentication checks at the web server and should be permitted access. It prevents computers from the 'outside' to have knowledge about the network on the 'inside' of the datacenter and it authorises the NetScaler Gateway ICA Proxy to set up a connection from the 'outside' to the 'inside'. It basically specifies where an outbound connection can connect to on the 'inside'.

Once a user launches a resource, externally (or internally for that matter) through NetScaler Gateway, at one point a secure ticket will be requested. As we will see shortly the STA ticket will eventually end up in the launch.ica file generated by StoreFront and/or Web-Interface.

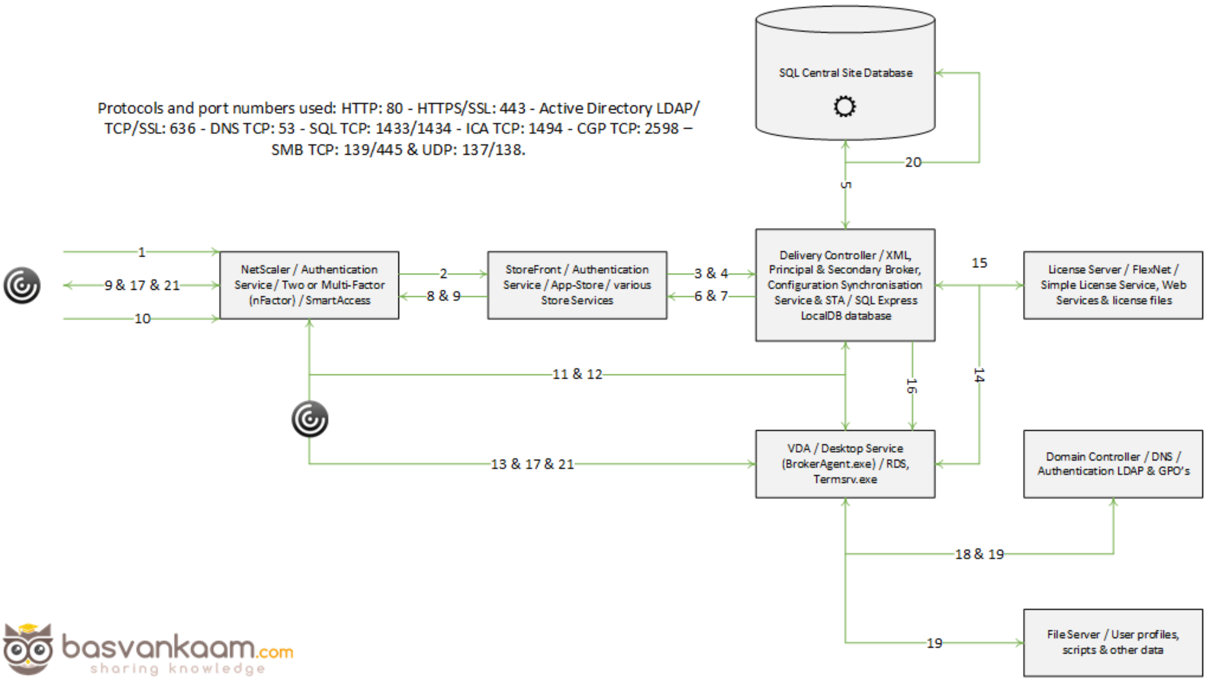
Once generated, the Delivery Controller hosting the STA service will hold the STA ticket information in memory for a configurable amount of time. As soon as a secure session is established the NetScaler Gateway responsible for handling the session only has to check the STA ticket (as part of the .ica launch file) with the STA service that originally generated the ticket. It (the STA service) does this from memory where the ticket was stored after it was created and send back to the StoreFront server as part of the XML formatted file mentioned earlier.

> **FMA fact:** The STA is only used when traffic traverses a NetScaler, so you don't have to worry about the STA service and its tickets when authentication takes place internally through StoreFront, for example.

1. Assuming that the login, authentication and enumeration process finished without any issues (see above) the user is now free to subscribe to and launch any applications and/or desktops that might have been assigned to him or her. As an example, let's assume that the user wants to launch a (XenApp) Hosted Shared Desktop session a.k.a. a published desktop.

2. After the user clicks the icon the launch request is send to the NetScaler Gateway from where it will be forwarded to the StoreFront server (1 & 2) see image below.
3. The StoreFront server will contact the Broker (XML/STA) service, or Delivery Controller, to find out if and where the resource is available and where it can be best started (3). This is where the well-known XenApp load balancing mechanism comes into play. Which as of XenApp 7.x needs to be configured through policies (or use the defaults).
4. During this time, the StoreFront server will also request an STA ticket from the Broker (XML/STA) service (4). It will include the user, domain and resource name it wants to start. It will also request a 'least loaded' server as part of the load balancing process.
5. The Broker (XML/STA) service will query the Central Site Database (ports Nr. 1433 and 1434) to find out which server can offer the requested resource (5), which is also referred to as the current Farm state. The Delivery Controller will than use this information together with its load balance algorithm to decide which server to connect to.
6. At this time, the Broker (XML/STA) service will create the STA ticket mentioned earlier. This will include information on the server and resource to connect to, amongst other information as discovered in the previous steps mentioned.
7. Next, the Broker (XML/STA) service will send this information back to the StoreFront server in the form of an XML formatted file (6 & 7).
8. Based on this information the StoreFront server will generate a launch.ica file (it uses the default.ica file as a template) containing the STA ticket and a whole bunch of other connection properties that are, or might, be needed (8). This will also include the FQDN/DNS name of the NetScaler Gateway itself.
9. StoreFront passes on this information down through the NetScaler Gateway onto the locally installed Receiver (9) which initiated the connection to begin with.
10. The locally installed Receiver will read and auto launch the launch.ica file to set up a connection to the NetScaler Gateway over 443 / SSL (10).
11. From here the NetScaler Gateway will first contact the Broker (XML/STA) service (this address is configured on the NetScaler as well) to verify if the earlier generated STA ticket, as part of the launch.ica file is still valid (11).
12. The Broker (STA) service will validate the STA ticket from memory. Once verified it will send back the IP address, port Nr. Resource name etc. of the machine and the resource it needs to connect to (12). Once done the STA ticket will be deleted.
13. The NetScaler Gateway will set up a new ICA connection using port 1494 (ICA) or 2598 (CGP – Common Gateway Protocol) depending on its configuration (13). Soon to include 'HDX Enlightened Data Transport' I'm sure.
14. The VDA will verify its license file with the, or a Delivery Controller (14).
15. The Delivery Controller checks with the Citrix License server to verify that the end user has a valid ticket (15). This will also be done for the Microsoft (CAL) licenses, regarding the Hosted Shared Desktop session and any published applications that might be involved.
16. Now, any applicable Citrix policies will be passed onto the VDA applying them to the session (16).
17. The Hosted Shared Desktop session is launched and the NetScaler Gateway acts as a proxy between the user and the XenDesktop resource in the data center (17).

18. User (Windows) authentication takes place between the domain controller and the Citrix Worker / Session Host (18).
19. The Citrix session will initialize; the Windows welcome screen appears. At this point the user profile is loaded, Group Policies (GPO's) are applied, scripts will be executed, drive and printer mappings are established and so on (19).
20. Somewhere in between the session/connection information will be passed on and registered in the Central Site Database where it will be used for future load balance purposes (20).
21. And finally, the Hosted Shared Desktop will be fully launched (21).


SQL Central Site Database
Protocols and port numbers used: HTTP: 80 - HTTPS/SSL: 443 - Active Directory LDAP/ TCP/SSL: 636 - DNS TCP: 53 - SQL TCP: 1433/1434 - ICA TCP: 1494 - CGP TCP: 2598 – SMB TCP: 139/445 & UDP: 137/138.
20
5
1
9 & 17 & 21
10
NetScaler / Authentication Service / Two or Multi-Factor (nFactor) / SmartAccess
2
8 & 9
StoreFront / Authentication Service / App-Store / various Store Services
3 & 4
6 & 7
Delivery Controller / XML, Principal & Secondary Broker, Configuration Synchronisation Service & STA / SQL Express LocalDB database
15
License Server / FlexNet / Simple License Service, Web Services & license files
11 & 12
16
14
13 & 17 & 21
VDA / Desktop Service (BrokerAgent.exe) / RDS, Termsrv.exe
Domain Controller / DNS / Authentication LDAP & GPO's
18 & 19
19
File Server / User profiles, scripts & other data
basvankaam.com
sharing knowledge

> **FMA fact**: The STA ticket gets generated and sent back after a user launches an application/desktop, and not during the resource enumeration process. It also includes information on the resource to be launched, including the server to launch the application on (load balance).

Internal user authentication through StoreFront

What happens when a user authenticates internally, directly to StoreFront? As you will see, besides the NetScaler in between both methods look very similar. Let's have a look. Same rules apply here, use port Nr. 443 where you can.

1. A user opens a web browser and connects to the internal StoreFront URL where he or she will fill in his or her username and password. This method is also referred to as Receiver for web sites as mentioned above (don't confuse this with the HTML 5 based Receiver for web, they're not the same). A locally installed Citrix Receiver can also be used to establish a direct connection to StoreFront, which is probably the preferred

method whenever possible. The earlier mentioned (NetScaler) Beacon functionality applies here as well.

2. Next the StoreFront authentication service will pick up the user credentials and contact a domain controller to authenticate the user in Active Directory over TCP port Nr. 389. Here I'd like to note that if domain pass-through authentication is enabled on the StoreFront server, this step would automatically be skipped. This is where the authentication token is built/generated — by default the StoreFront Authentication Service will take care of this. However, as of StoreFront version 3.0, Citrix re-introduced XML-based user authentication. By simply running a few PowerShell scripts user authentication falls back to the XenDesktop/XenApp XML service, which is equal to how Web Interface used to handle things. Particularly useful when StoreFront is not in the same domain as XenDesktop / XenApp and when it is not possible to set up an Active Directory trust, or multiple. Just be aware that this method will be disabled by default. As of StoreFront version 3.5 and upwards PowerShell is no longer needed to enable XML based user authentication, it can be enabled and disabled directly from the StoreFront management console.
3. Once authenticated the user credentials will be forwarded, as part of a XML query, to the configured Broker (XML) service on one of the available Delivery Controllers. Both these transactions will use port Nr. 80 by default, which of course can be changed to 443 (SSL).
4. In between, StoreFront will check its local data store for any existing recourse subscriptions and stores these in memory.
5. During the next phase the Broker (XML) service will again contact a domain controller (default over port Nr. 389) to validate the user credentials, note that this is different to the user authentication process, as we've established earlier. During this process, it will find out to which security groups (SID's) the user belongs.
6. Here you basically authenticate/validate against LDAP two times:
7. Through Storefront, either using the SF Authentication Service or via SF to the XML Service on one of your Delivery Controllers -> Active Directory, this will, as mentioned earlier generate/built the authentication token.
8. Through the XML Service (validation) -> Active Directory, to find out the accompanying security group SID's used for resource enumeration.
9. With this information the Delivery Controller, or Broker (XML) service, will contact the central Site Database to find out which resources have been assigned to the user. It does this over port Nr. 1433 / 1434.
10. This data will then be gathered and send back to the StoreFront server in the form of an XML formatted file, through/using the Broker (XML) service.
11. Based on this information StoreFront will generate a web page containing all the assigned resources, which will be presented to the user. The users home screen will be populated with any pre-subscribed resources (Keywords). Depending on how you connected, your resources will be displayed either directly using a Receiver for web sites or you'll find them within the locally installed Citrix Receiver instead. The user will be able to browse its own personal app store for any assigned resources to which he or she can subscribe and then launch.

The (internal) launch process

Now that we've seen which steps are involved when launching a resource externally, a Hosted Shared Desktop in this case, let's have a look and see what happens when we launch a pooled VDI virtual machine internally. Since we do not have to 'deal with a NetScaler, we won't have to worry about the STA as well. After this we will have looked at an external and internal resource launch, a HSD, which is comparable to a published application, and a VDI virtual machine. Again, user authentication and resource enumeration has successfully completed, here we go (again).

1. Assuming the login, authentication and enumeration process finished without any issues (see the above sections) the user is now free to subscribe to and launch any applications and/or desktops that might have been assigned to him or her. As mentioned, this time we will launch a pooled VDI virtual machine. I'll assume that the VM is pre-subscribed and already present on the user's home screen, never mind how we connected: locally installed Receiver or using the Receiver for web sites.
2. After the user clicks the icon the StoreFront server will contact the Broker (XML) service on the Delivery Controller, to check if any registered VDA's are available. It does this by communicating with underlying Hypervisor platform (your Host Connection) through the Host service on the Delivery Controller.
3. If needed it will first start / boot a VM. It's not uncommon to pre-boot a few VM's, since, as you can probably imagine, this will positively influence the overall user experience.
4. Next the Delivery Controller, or Broker (XML/STA) service, will contact one of the VDA's and sends a startlistening request. By default, the VDA isn't listening for any new connections on port Nr. 1494 or 2595 until it gets notified that a user wants to connect.
5. As soon as the VDA is listening, the Broker (XML/STA) service will send this information back to the StoreFront server in the form of an XML formatted file (it isn't an actual XML file).
6. Based on this information the StoreFront server will then generate a launch.ica file (it uses the default.ica file as a template) containing the IP address of the VDA and a whole bunch of other connection properties that are, or might, be needed. This is send down to the user, or better said, the Citrix Receiver.
7. The locally installed Receiver (or HTML 5 based Receiver) will read and autolaunch the launch.ica file initiating a direct connection from the users end-point to the VDA using the ICA protocol over port 1494.
8. At the same time, the installed VDA will verify its license file with the Delivery Controller.
9. The Delivery Controller checks with the Citrix License server to verify that the end user has a valid ticket.
10. Now, any applicable Citrix policies will be passed onto the VDA applying them to the session.
11. User (Windows) authentication takes place between the domain controller and the Citrix VDI VM.
12. The Citrix session will initialize; the Windows welcome screen appears. At this point the user profile is loaded, Group Policies (GPO's) are applied, scripts will be executed, drive and printer mappings are established and so on (19).

13. Somewhere in between the session/connection information will be passed on and registered in the Central Site Database.
14. And finally, the VDI session is fully launched.

Site policies

When we publish resources, either hosted shared desktops, VDI-based virtual machines or published applications, we normally would use a combination of Catalogs and Delivery Groups to grant or allow access to these resources. Although this works fine in most cases, using PowerShell we can get a bit more granular.

Entitlement policies – These apply to pooled and shared desktops. With entitlement policies you can explicitly deny a certain user from a group of users' access to a pooled and/or shared desktop. Let's say you have a group of 50 users and you want to exclude five users, using these policy rules you won't have to create a separate group of users to exclude, you can just exclude those five users without affecting any of the other users. There are two types of Entitlement policies:

1. BrokerEntitlementPolicyRule: this one is issued for access to desktops.
2. BrokerAppEntitlementPolicyRule: this one is used to control access to applications.

Assignment policies – These basically do the same thing as the Entitlement policies described above, only they apply to dedicated private desktops. Again, there are two policies:

1. BrokerAssignmentPolicyRule: this one is issued for access to desktops.
2. BrokerAppAssignmentPolicyRule: this one is used to control access to applications.

Before we move on to the Site Access policy I'd like to point out another Entitlement policy gotcha: Once we've configured a Delivery group with the Desktops and Applications delivery type we can use PowerShell to limit access to the HSD. Let me explain what I mean here.

By default, when you create a delivery group with the delivery type set to Desktops and Applications, Studio creates one Desktop Entitlement Policy Rule and one App Entitlement Policy Rule for the group, meaning that each user is entitled to one desktop session and one app session. Studio doesn't expose the user filter on these objects, so both are available to all users of the delivery group.

Using the PowerShell command: Set-BrokerEntitlementPolicyRule we can change this behaviour.

It can set the IncludeUserFilterEnabled parameter to True instead of False, enabling the user filter, and it also lets you add an AD security group, this way limiting access to just that group and that group alone, as opposed to all users who are members of the Delivery group.

Site Access policies

Site Access policies – This isn't directly about the users connecting, it is more about connections in general and the conditions that need to be met once a connection gets established: things like client IP addresses, the protocol used, Smart Access filters, hostnames etc. Based on this information, connections can be excluded or denied access as well.

> **FMA fact**: When a Delivery Group gets created, two access rules are created and added by default, one for direct connections and one for connections through NetScaler. Using PowerShell we can look at and change these access rules, as we see fit.

To see what your Site Access policies currently look like, open PowerShell and type:

Get-BrokerAccessPolicyRule followed by the -DesktopGroupName command, so that you won't get overloaded with all Access policies currently enabled.

To edit these policies you will use the Set-BrokerAccessPolicyRule command.

The Windows authentication process

As I am primarily focusing on the Citrix side of things here I deliberately left out the Windows / Domain Authentication process. However, you do need to be aware that every time you initiate a fresh Citrix session, by launching a hosted shared desktop session, for example, or by starting published applications, a few things happen in the background from a Windows perspective as well. When you launch a resource you are basically logging in on the server machine where that specific application or desktop is being published. As a result, you will be logged on as a normal Windows / domain user before your resource will actually start. The steps involved in the background do not differ when compared to logging into a 'normal' Windows server or desktop machine. These steps include:

- User logon and authentication
- Profile load
- GPO processing
- Startup scripts
- Drive mapping
- Printer mappings

Receiver and HTML5

From my previous examples I assume that you already have the Citrix Receiver installed locally, which is a pretty common scenario. But if you don't, you have a few options. First (we already covered some if this earlier) when you connect to StoreFront, either directly or through the NetScaler Gateway as we've talked about, it automatically checks if there is a Receiver installed locally and which Operating System you are running. If not, it will guide you to a download section or page (usually Citrix's) where you will be able to download the Citrix Receiver. There are a few ways administrators can implement this.

HTML 5 receiver to the rescue

If for whatever reason you are unable or not allowed to install a Citrix Receiver locally, Citrix offers the Receiver for HTML 5. You will still be able to connect to StoreFront / NetScaler and launch your resources without any loss of functionality.

Although not enabled by default, StoreFront has a build-in HTML 5 based Receiver, which will kick in at launch time. It does this by fetching the HTML 5 engine from the StoreFront and making it part of the local browser.

Note that you must use a HTML 5 supported browser for this to work. Basically, your browser becomes your Receiver handling the launch.ica file. When you close the browser, you close the session. Even when your users will have Receiver installed you can enable it anyway as it will function as a fallback mechanism.

Broker, XML, STA and Principal

Be aware that the STA (service) is also part of the Broker service, and has been as of Presentation Server 4.0. Before that it was written as an ISAPI extension for Microsoft Internet Information Services, or IIS. I also highlighted the so-called XML service multiple times. I put the XML and STA services between brackets because as of XenDesktop 4.x the XML service (ctxxmlss.exe) has been rewritten in .NET and became part of the Broker service.

The Broker service is build up out of three separate services, all handling different tasks, it brokers connections, it enumerates resources and it acts as the Secure Ticket Authority, generating and validating STA tickets. With the re-introduction of the LHC the Broker services is now also knows as the Principal Broker Service. Every two minutes the (Principal) Broker Service will be checked for configuration changes. If a configuration change has been detected it will be copied over, or synchronised to the High Availability Service/Secondary Broker Service.

> **FMA fact:** Make sure that the Broker (XML/STA) service on the NetScaler and the Storefront server is configured identically. The same applies to the load balance/fail over order in which you configure them.

Key takeaways

- There are two main authentication points within a Flex Management-based Architecture: NetScaler (optional) and StoreFront.
- Knowing the difference between the IMA and the FMA, how traffic flows throughout each component, and the way they are supposed to interact is or can be vital to successfully troubleshooting the FMA.
- As of version 3.0, StoreFront can also use the XML service for authenticating users.
- Note that there is a distinct difference between authentication and verification. Authentication is to make sure that somebody is who he or she claims to be. Verification is done to find out which resources are assigned (permissions) to the user, which will then be displayed in the user's store, ready for subscription.
- User authentication and resource enumeration basically go hand-in-hand.
- The STA only applies when connections are coming in externally through NetScaler.
- The STA service is part of the Broker server, and so is the perhaps better-known XML service.
- The HTML5-based Citrix Receiver, as part of your Internet browser, can offer the exact same functionality and features as a natively installed Receiver.
- The Windows authentication process is also involved when launching a Citrix published resource.
- Site policies allow us to exclude certain users or to apply certain policies when specific conditions are met. PowerShell can be used to manage and configure Site policies.