

¿Por que Securizar tu Tráfico entre XML Brocker y WebInterface/CloudGateway?

Escrito por cristiansasn

Martes, 16 de Abril de 2013 08:51 - Actualizado Martes, 16 de Abril de 2013 08:52

El servidor Citrix XML es un componente de Citrix XenApp y XenDesktop que se utiliza para enumerar y proporcionar entradas seguras para los usuarios que utilizan a la WebInterface o CloudGateway.

El Servicio XML Citrix es un servicio de Windows que forma parte del Citrix XenApp y XenDesktop.

Se utiliza para proporcionar datos enviados tras solicitudes a los componentes XML de Citrix. Esta tecnología se introduce con MetaFrame 1.8 SP2. Hasta XenApp 6.0 cada servidor XenApp puede ser un servidor XML. En XenApp 6.5 se introduce la arquitectura de controlador de trabajo y sólo un servidor con la función de controlador puede ser un server XML. Un servidor con la función de controlador es responsable de la gestión de las explotaciones.

El servicio XML de Citrix es más comúnmente utilizado para proporcionar a los usuarios acceso a sus aplicaciones y escritorios a través de un portal web, aprovechando los servicios de WebInterface, CloudGateway (Storefront) o Access Gateway.

Se recomienda tener varios servidores Citrix XML, lo más cerca posible de los DataCollector (servidores de zona XenApp) y Desktop Delivery Controller (XenDesktop).

El protocolo XML de WebInterface y/o StoreFront utiliza texto plano en el envío de datos. Ello nos hace **potencialmente** posibles **victimias** para un ataque realizado internamente.

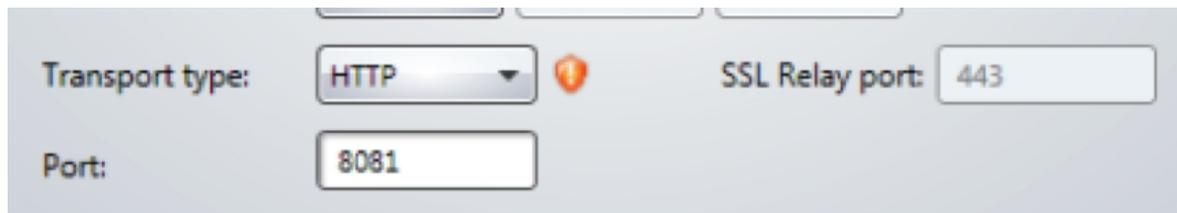
¿Seguridad?

¿Por que Securitizar tu Tráfico entre XML Brocker y WebInterface/CloudGateway?

Escrito por cristiansasn

Martes, 16 de Abril de 2013 08:51 - Actualizado Martes, 16 de Abril de 2013 08:52

Mi entorno de laboratorio será la mesa de pruebas para esta demostración. Mi actual granja utiliza el puerto 8081 para el transporte XML. No utilizamos ningún mecanismo de encriptación y securización del tráfico.



The image shows a configuration panel with the following fields:

- Transport type: HTTP (dropdown menu)
- Port: 8081 (text input)
- SSL Relay port: 443 (text input)

La prueba a realizar es sencilla. Vamos a realizar los siguientes pasos sobre un entorno de XenApp con CloudGateway.

1. Abrir URL CloudGateway
2. Validar-nos en StoreFront
3. Lanzar aplicación

Durante este proceso vamos a lanzar un snifer de red, capturando todo el tráfico.

Conectamos en el entorno y nos validamos.



The image shows the Citrix Receiver login interface with the following elements:

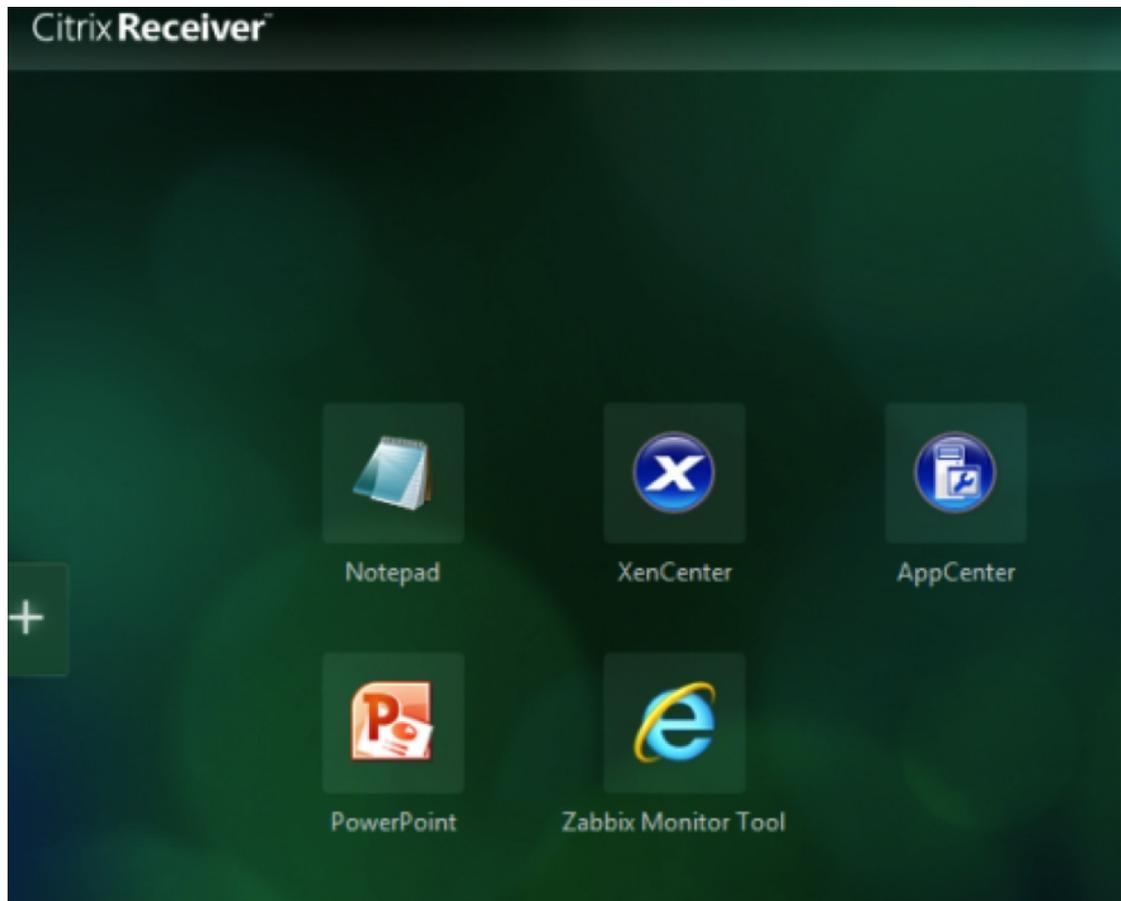
- Citrix Receiver logo
- Nombre de usuario: [text input]
- Contraseña: [password input]
- Iniciar sesión button

¿Por que Securizar tu Tráfico entre XML Brocker y WebInterface/CloudGateway?

Escrito por cristiansan

Martes, 16 de Abril de 2013 08:51 - Actualizado Martes, 16 de Abril de 2013 08:52

Se enumeran las aplicaciones disponibles.

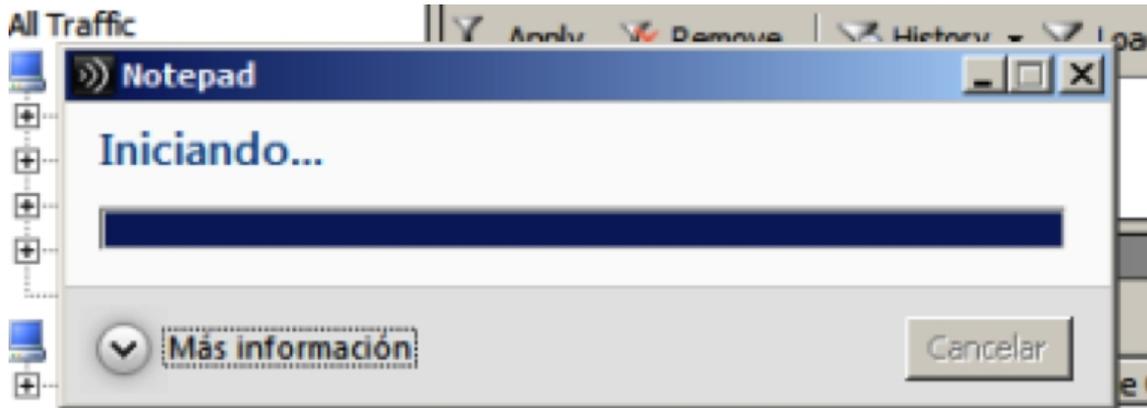


Lanzamos una aplicación.

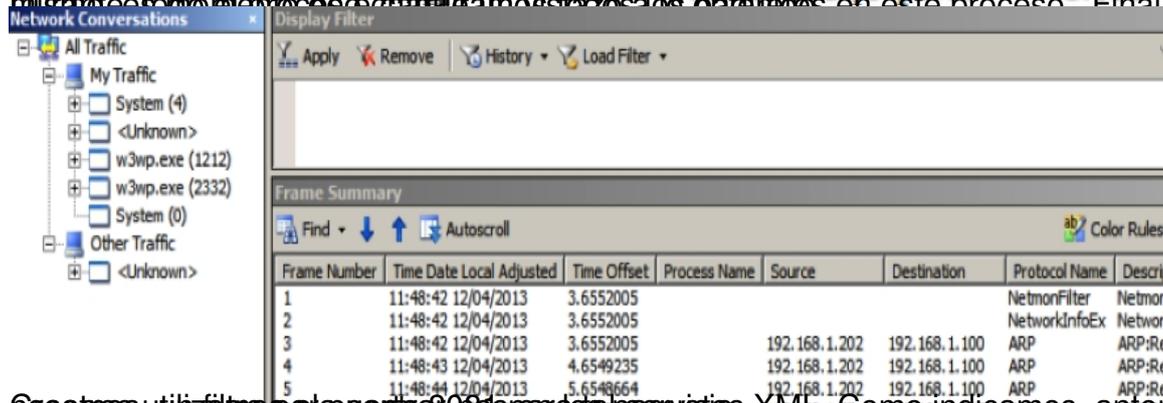
¿Por que Securizar tu Tráfico entre XML Broker y WebInterface/CloudGateway?

Escrito por cristiansan

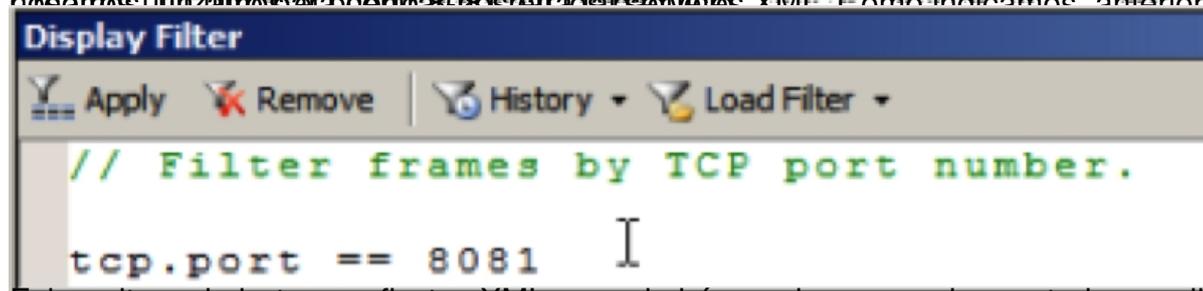
Martes, 16 de Abril de 2013 08:51 - Actualizado Martes, 16 de Abril de 2013 08:52



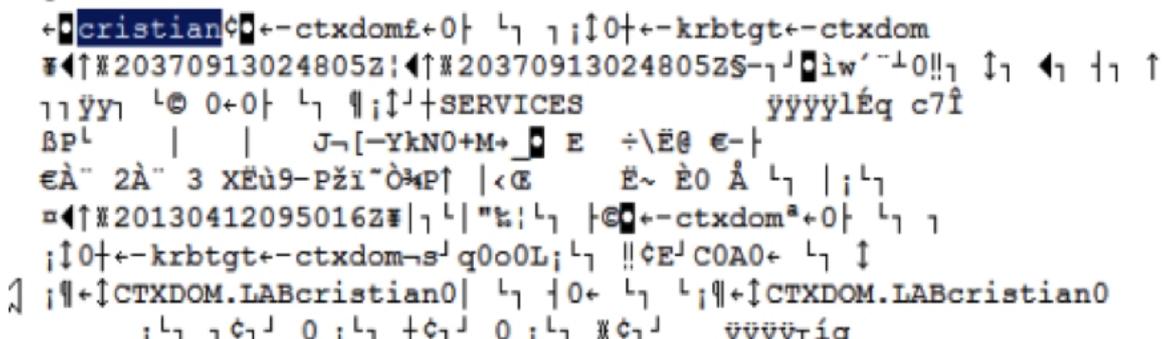
El primer paso es el de configurar la red de destino de los paquetes de obtención de datos. Finalizado el



Como se utilizó el filtro de paquetes de protocolo XML. Como indicamos anteriormente



El primer paso es la configuración de XML. Como se hizo la búsqueda para poder ver todo aquello que



El primer paso es de configurar el servicio de configuración en el servidor de CloudGateway de los

¿Por que Securizar tu Tráfico entre XML Brocker y WebInterface/CloudGateway?

Escrito por cristiansan

Martes, 16 de Abril de 2013 08:51 - Actualizado Martes, 16 de Abril de 2013 08:52

The image shows a Wireshark packet capture analysis. The top section is the 'Frame Summary' table, which lists the following information for frame 190: Time Date Local Adjusted: 11:49:33 12/04/2013, Time Offset: 54.7130421, Process Name: System, Source: 192.168.1.55, Destination: 192.168.1.51, Protocol Name: HTTP, and Description: HTTP:Reque. Below this is the 'Frame Details' pane, which shows the following information: Frame: Number = 190, Captured Frame Length = ... Ethernet: Etype = Internet IP (IPv4), Destinat... Ipv4: Src = 192.168.1.55, Dest = 192.168.1.51 Tcp: Flags=...AP..., SrcPort=10758, DstPort=H... SrcPort: 10758 DstPort: HTTP(80) SequenceNumber: 206956800 (0xC55E900) AcknowledgementNumber: 2544197535 (0x97A55F9F) DataOffset: 80 (0x50) Flags: ...AP... Window: 261 (scale factor 0x8) = 66816 Checksum: 0x35FF, Good UrgentPointer: 0 (0x0) TCPPayload: SourcePort = 10758, Destination... The 'Hex Details' pane shows the raw hex data of the frame, with the following values: 0320 6E 49 64 3D 34 6C 74 70 0328 6C 6C 6E 75 74 67 7A 6D 0330 66 6E 35 35 33 7A 75 72 0338 32 75 34 35 0D 0A 0D 0A 0340 75 73 65 72 6E 61 6D 65 0348 3D 63 72 69 73 74 69 61 0350 6E 26 70 61 73 73 77 6F 0358 72 64 3D 45 6E 65 72 6F 0360 32 30 30 39 26 62 75 74 0368 74 6F 6E 4C 61 62 65 6C 0370 3D 49 6E 69 63 69 61 72 0378 2B 73 65 73 69 25 43 33 0380 25 42 33 6E 26 73 74 61 0388 74 65 43 6F 6E 74 65 78 0390 74 3D

Csrf-Token: B099922AA5FFCD2BBBA0176481CED7A2
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.57
Safari/537.17
Referer: http://192.168.1.51/Citrix/internalWeb/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-ES, es; q=0.8
Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.3
Cookie: CsrfToken=B099922AA5FFCD2BBBA0176481CED7A2;
CtxsPluginAssistantState=Done; ASP.NET_SessionId=
4ltp1lnutzgzmfn553zur2u45

username=cristian&password=[REDACTED]&buttonLabel=Iniciar+sesi%

C3&B3n&stateContext=[REDACTED] 6 6 ^â°ôä0T-[-