Windows Server Update Services

rcoil / 2019-10-22 09:15:12 / 浏览数 3247 安全技术 技术讨论 顶(0) 踩(0)

0x00 前言

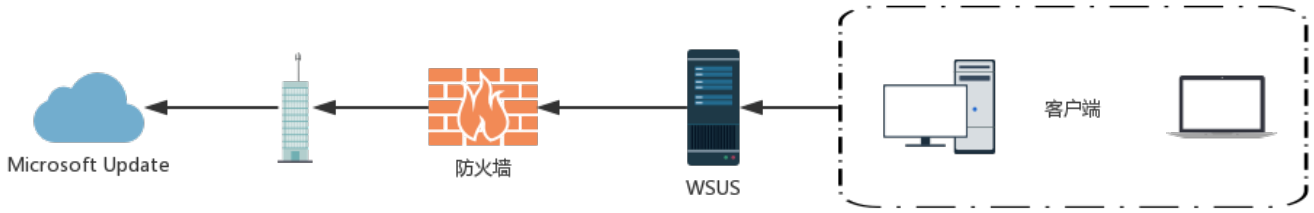
为了让 Windows 系统和其他 Microsoft 产品能够更安全、更稳定，因此 Microsoft
会不定期在其网站上推出最新的更新程序供用户下载与安装，而用户可以通过以下方式来取得这些程序：

- 手动连接 Microsoft update 网站
- 通过 Windows系统的自动更新功能

然而以上两种方式对企业内部来说，都可能会有以下缺点。

- 影响网络效率：如果企业内部每台计算机都自行上网更新，将会增加对外网络的负担。
- 与现有软件相互干扰：如果企业内部使用的软件与更新程序发生冲突，则用户自行下载与安装更新程序可能会影响该软件或更新程序的正常运行。

WSUS 是一个可以解决上述问题的产品，企业内部可以通过 WSUS 服务器集中从 Microsoft update
网站下载更新程序，并且在完成这些更新程序的测试工作，确定对企业内部计算机没有不良影响后，在通过网管审批程序，将程序部署到客户机上。本段文字简述来自
利用WSUS部署更新程序

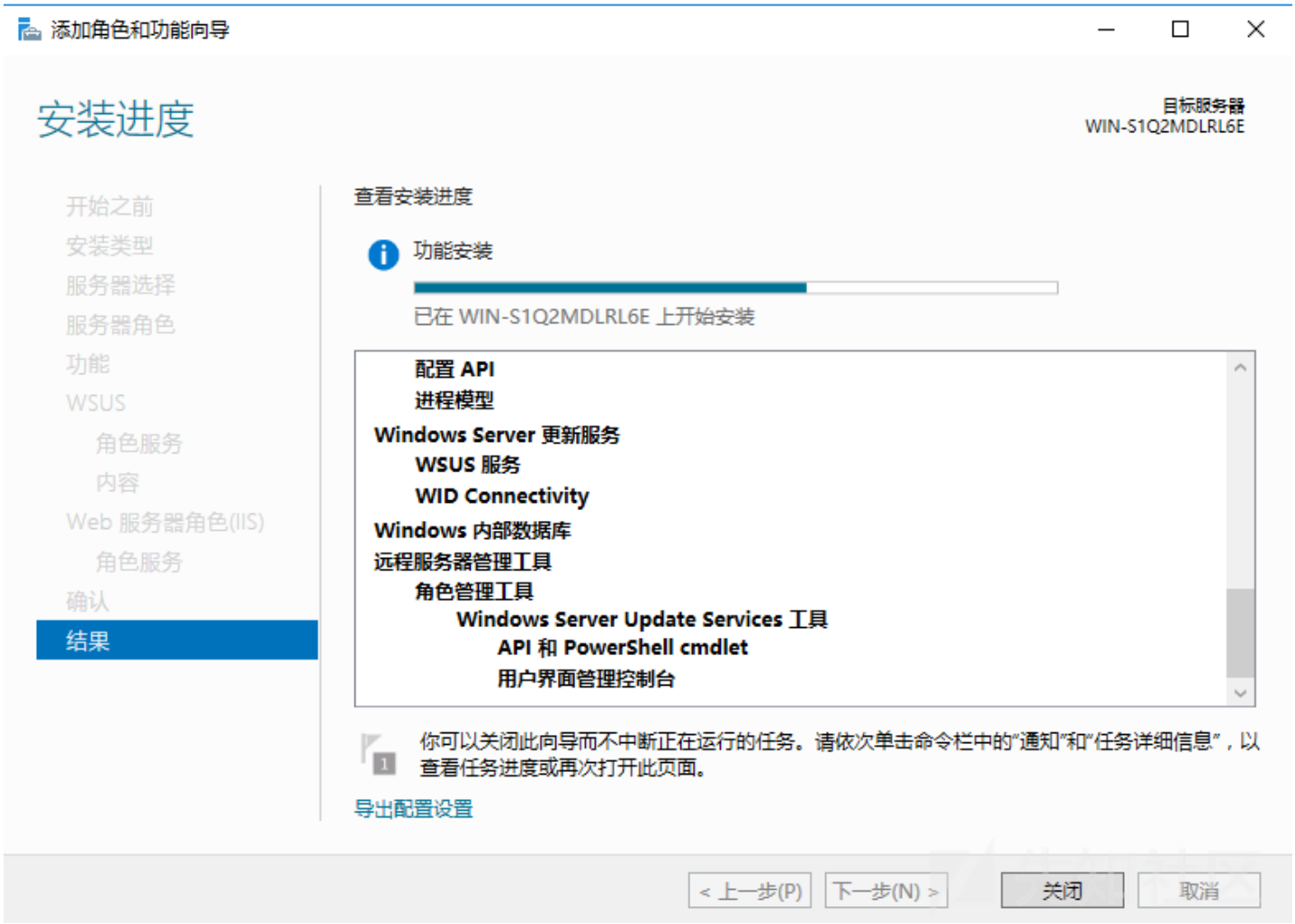


0x01 安装与配置 WSUS

1.1、安装

```
Sytem■Windows Server 2012 R2 Standard x64
Domain■ rcoil.me
```

一路点击默认即可。

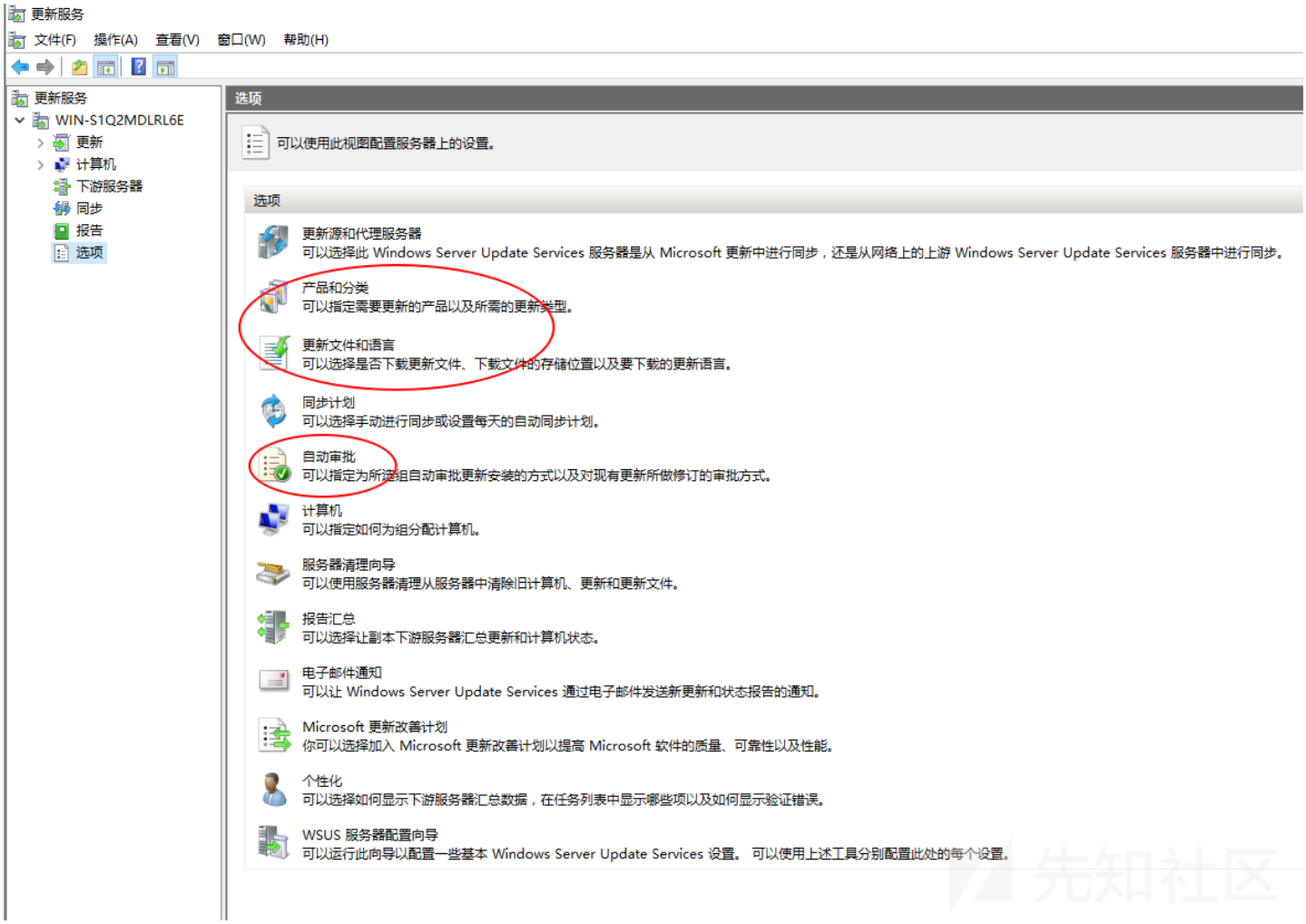


此时等待安装完成即可，后续选择 WSUS 配置向导继续配置即可，详细安装过程请参考以下链接。

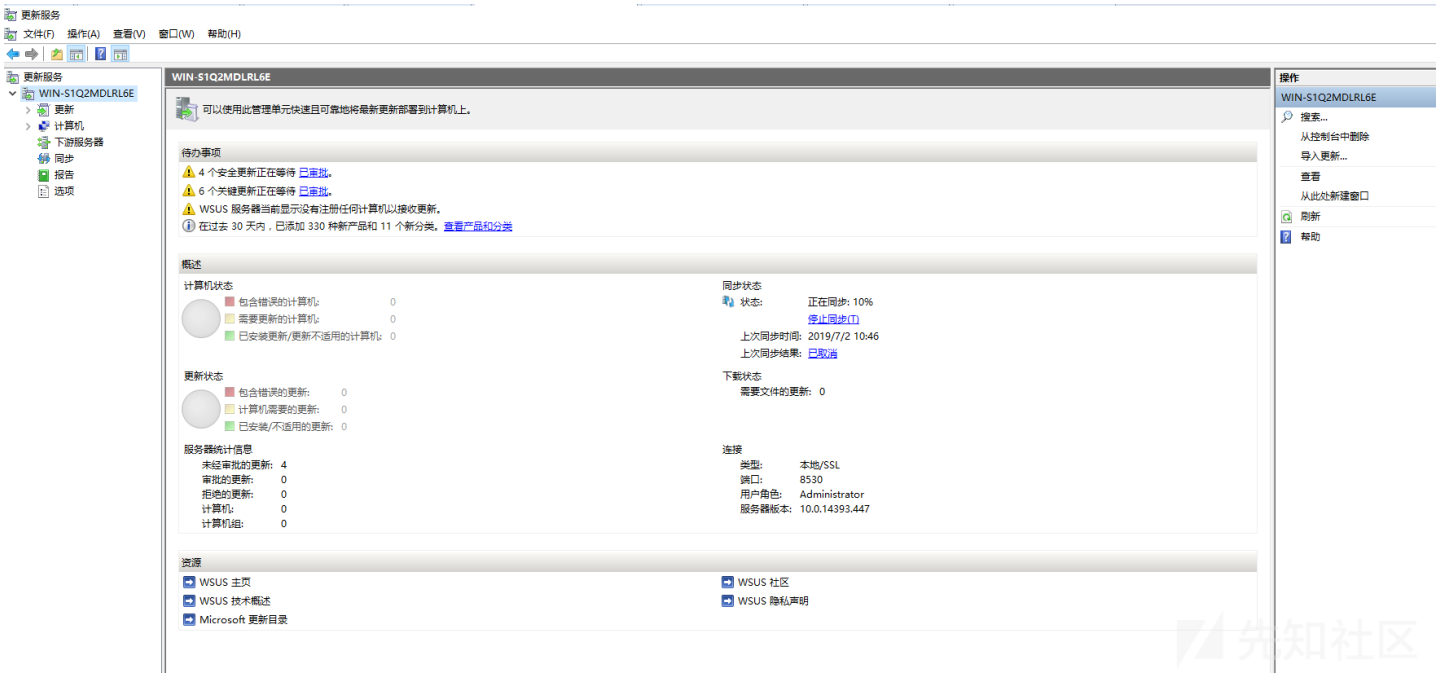
安装步骤参考：[wsus服务器搭建和客户端设置]

1.2、配置

打开更新服务，进行几个必要配置。



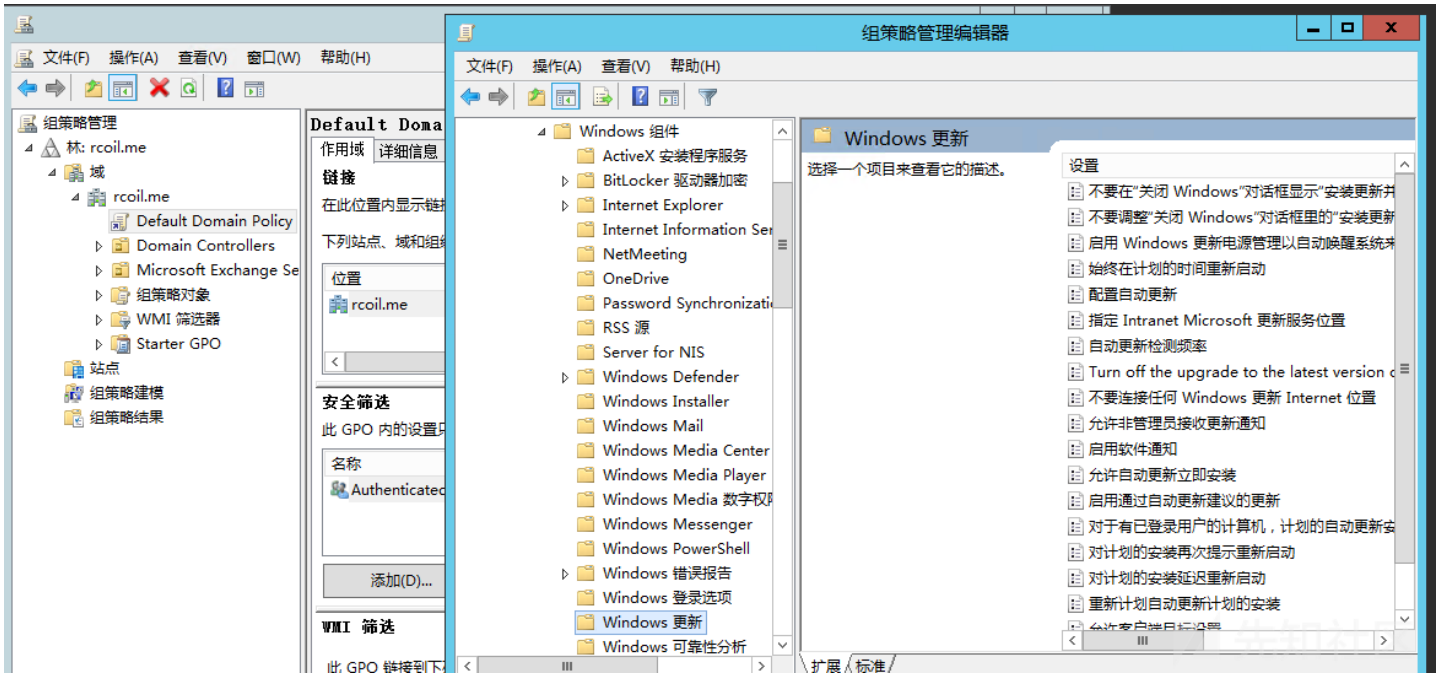
同步过程。



1.3、连接更新服务器

在 Default Domain Policy 做一个影响全域计算机的自动更新策略，也可以新建个 GPO。

■■1■■■■■■■■■■ (GPMC) ■■■■■■■■ Default Domain Policy■ GPO■■■■■“■■”■
■■2■■ GPMC ■■■■■■“■■■■■”—>“■■”—>“■■■■”—>“Windows ■■”—>“Windows ■■”■

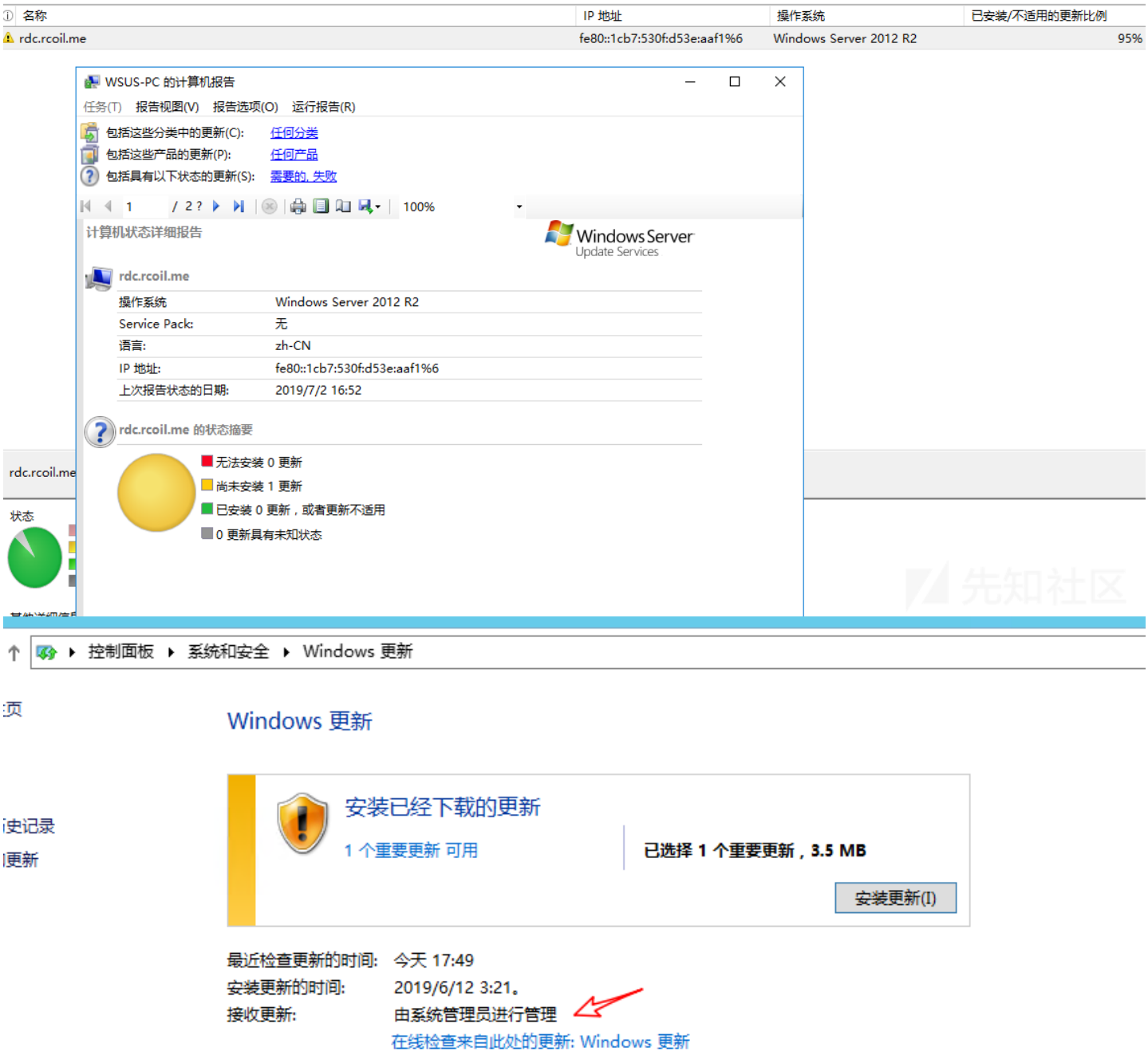


■■3■■■■■■■■■■■■"■■■■■■"■"■■ Intranet Microsoft ■■■■■■"■"■■■■■■■■■■"■"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"■"■■■■■■■■

■■4■gpupdate /force ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

1.4、WSUS查看状态报告

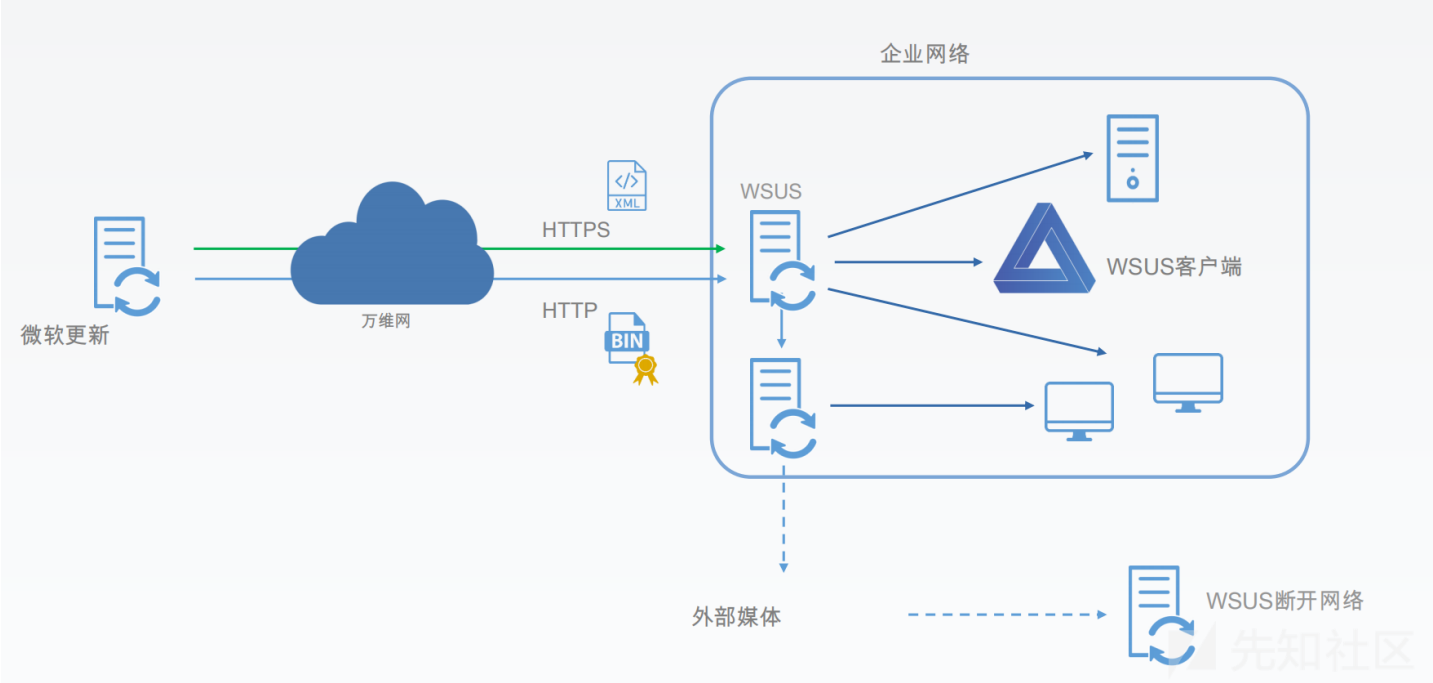
默认情况下，在 WSUS 控制台中是无法查看状态报告的，如果想正常的查看状态报告，需要一些插件和功能的支持，这个自行解决。



0x02 获取 WSUS 内部操作信息

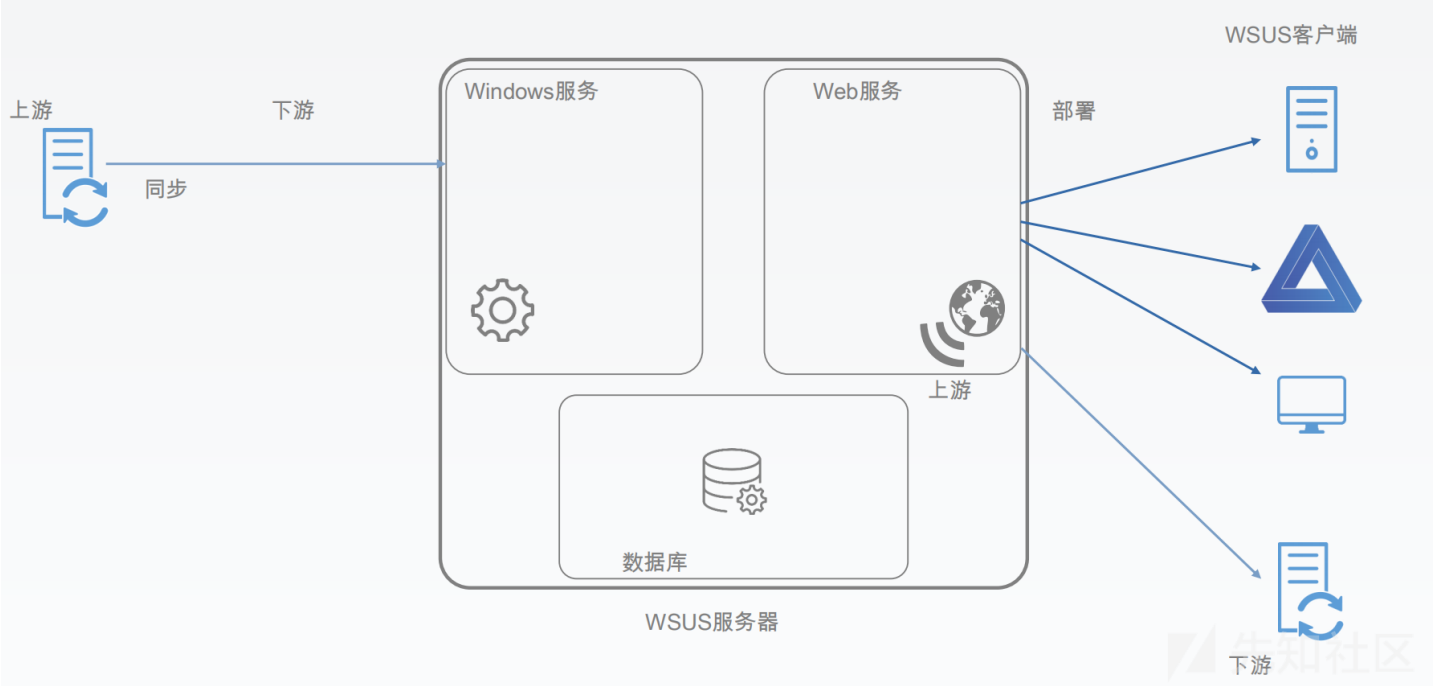
WSUS 由三个基本组成

- IIS Web服务（负责与客户端机器进行通信）
- 数据库 （存储各类元数据）
- 服务（提供更新服务及协调以上两者）
  ### 2.1、WSUS 架构

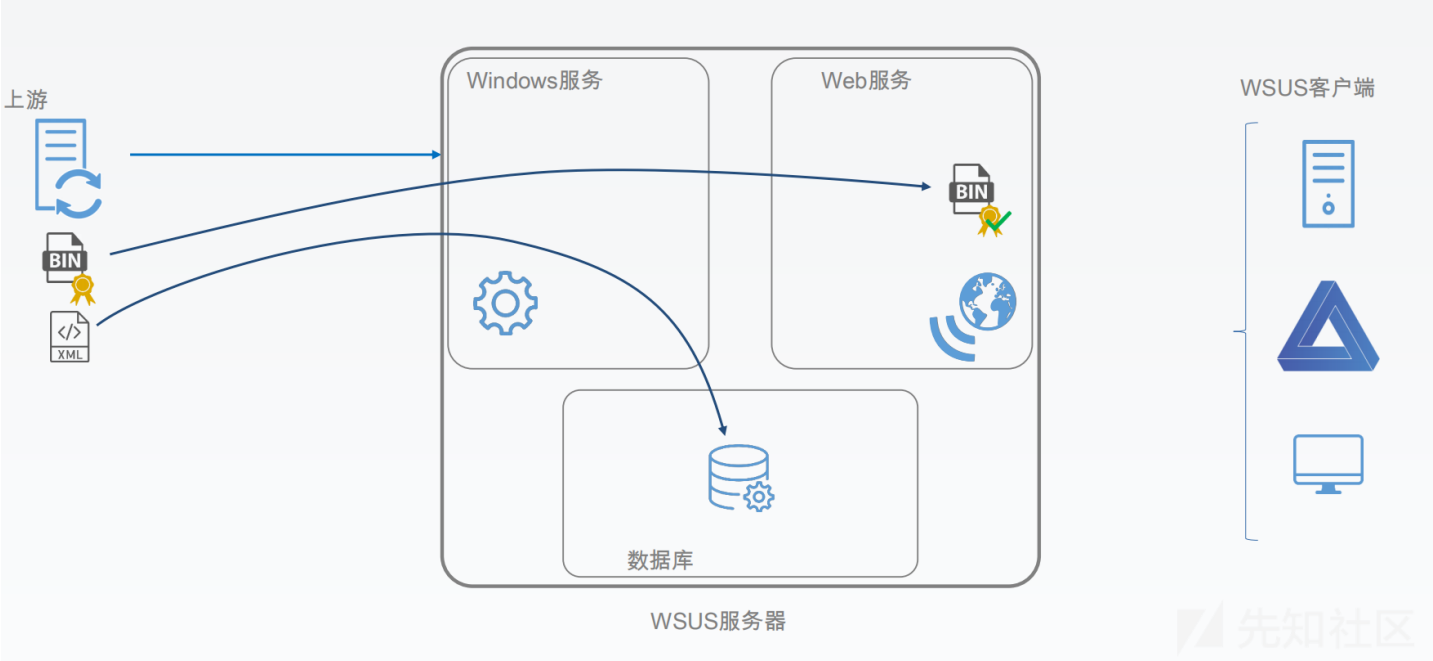


2.2、WSUS 服务器的组件(更新过程)

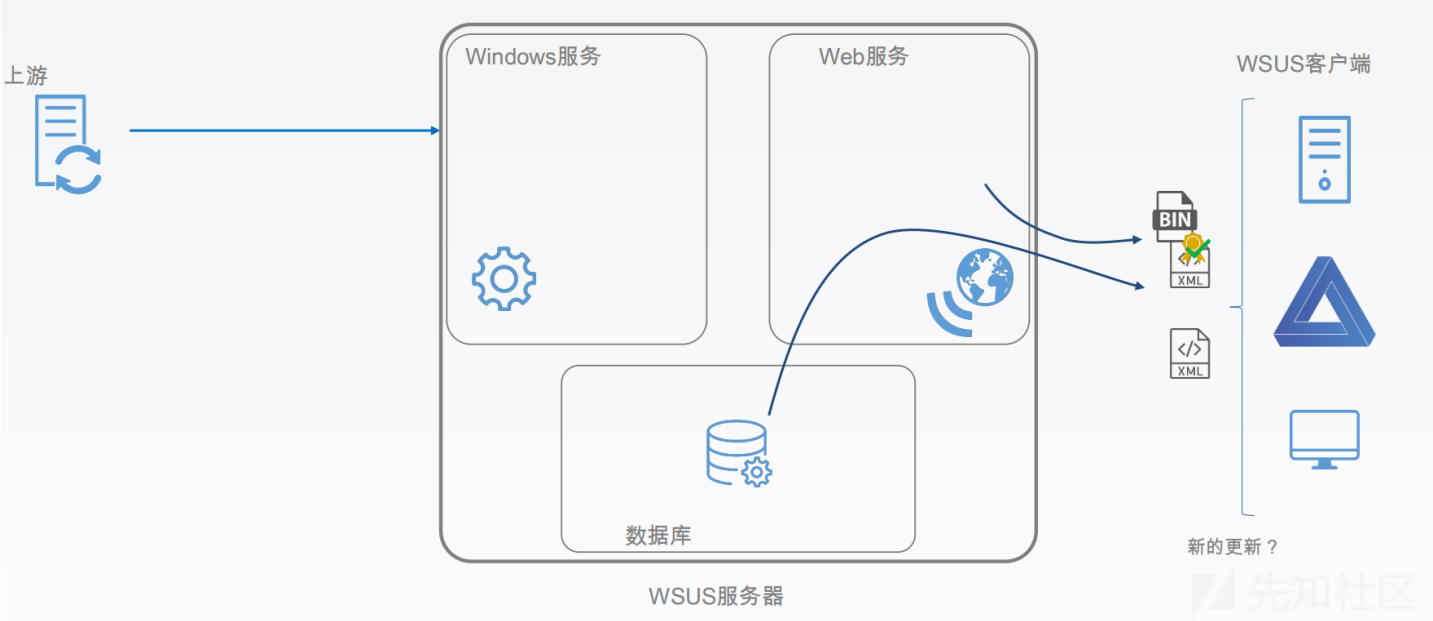
- 图1



- 图2



- 图3



2.3、 数据库

WSUS 在安装的时候提供了 2 种可选择的 数据库类型：SQL Server■■■和 WID。默认情况下使用
WID，该数据库仅用命名管道进行连接访问（文中有例子），该数据库也可以看作是一个轻量级的 SQL Server数据库，其中 SQL命令都是相同的。

该数据库包含了 WSUS
的元数据更新、部署日志、客户端机器信息、客户端配置信息等关系表。但是由于有统一的触发器对数据进行检测，所以插入的野生数据可能会被拒绝。

从注册表项中可以获取更新服务器地址、更新频率、提高非管理员等等等。

在客户端上确定 WSUS 地址

```
reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate
 reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU
```

```
beacon> shell reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate
[*] Tasked beacon to run: reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate
[+] host called home, sent: 95 bytes
[+] received output:

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate
    WUServer    REG_SZ    http://WSUS-PC:8530
    WUStatusServer    REG_SZ    http://WSUS-PC:8530

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate\AU

beacon> shell reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU
[*] Tasked beacon to run: reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU
[+] host called home, sent: 98 bytes
[+] received output:

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate\AU
    NoAutoUpdate    REG_DWORD    0x0
    AUOptions    REG_DWORD    0x3
    ScheduledInstallDay    REG_DWORD    0x0
    ScheduledInstallTime    REG_DWORD    0xf
    NoAutoRebootWithLoggedOnUsers    REG_DWORD    0x1
    AutoInstallMinorUpdates    REG_DWORD    0x1
    UseWUServer    REG_DWORD    0x1
    DetectionFrequencyEnabled    REG_DWORD    0x1
    DetectionFrequency    REG_DWORD    0x1
```


- IE 的代理情况

```
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"
```

- 获取数据库连接必要信息

```
reg query "HKLM\SOFTWARE\Microsoft\Update Services\Server\setup" /v SqlDatabaseName
 reg query "HKLM\SOFTWARE\Microsoft\Update Services\Server\setup" /v SqlServerName
```

在这里就遇到一个问题，在连接 Windows内部数据库的时出现无法连接的情况，使用管理员权限即可。

```
beacon> execute-assembly /Users/rcoil/Desktop/Github/CSharp_Tools/SharpWSUS/SharpWSUS/bin/Debug/SharpWSUS.exe
[*] Tasked beacon to run .NET program: SharpWSUS.exe
[+] host called home, sent: 115755 bytes
[+] received output:
Hello World!

[WSUS-PC] rcoil */7532
beacon>
```

```
            // 'Server=np:\\.\pipe\MICROSOFT##WID\tsql\query;Database=SUSDB;Integrated Security=True'
            sqlcQuery.ConnectionString = @"Server=np:\\.\pipe\" + clWSUSData.sDatabaseInstance + @"\tsql\query;Database=" + clWSUSData.sDatabaseName +
        }
        else
        {
            Console.WriteLine("We are checking for 2008, 2012, 2016.");
            return null;
        }

        try
        {
            sqlcQuery.Open();
            Console.WriteLine("Hello World!");
        }
        catch(Exception e)
        {
            Console.WriteLine(e);
```



```
sqlcmd.exe -S "np:\\.\pipe\MICROSOFT##SSEE\tsql\query" # 2008
sqlcmd.exe -S "np:\\.\pipe\MICROSOFT##WID\tsql\query" # 2012■■
```

```
```

- 获取 WSUS 数据库中的计算机信息

```
beacon> execute-assembly /Users/rcoil/Desktop/Github/CSharp_Tools/SharpWSUS/SharpWSUS/bin/Debug/SharpWSUS.exe
[*] Tasked beacon to run .NET program: SharpWSUS.exe
[+] host called home, sent: 120875 bytes
[+] received output:

Computers Name                IPAddress                   ClientVersion       OSDescription                ComputerMake        LastReportedStatusTime

rdc.rcoil.me                  fe80::1cb7:530f:d53e:aaf1%6 7.9.9600.19164                                    VMware, Inc.        2019/7/5 1:45:37
exchange2013.rcoil.me         192.10.20.199               7.9.9600.16384                                   VMware, Inc.        2019/7/5 1:52:22
wsus-pc.rcoil.me              fe80::3c00:ce41:fc7f:3211%6 10.0.14393.351      Windows Server 2016 Datacenter VMware, Inc.        2019/7/5 1:14:24
desktop-s7e2bep.rcoil.me      192.10.20.107               10.0.17134.799      Windows 10 Pro               VMware, Inc.        2019/7/3 6:32:43
prdc.rcoil.me                 192.10.20.232               7.9.9600.16384                                   VMware, Inc.        2019/7/5 1:50:19


[WSUS-PC] rcoil */7480
beacon>
```

```
            // 这样的查询方式偏慢
            SqlDataReader sqldrReader;
            sqlCommFun.CommandText = "exec spGetAllComputers";
            try
            {
                sqldrReader = sqlCommFun.ExecuteReader();
                int count = sqldrReader.FieldCount;

                while (sqldrReader.Read())
                {
                    Console.WriteLine("{0,-30} {1,-30} {2,-20} {3,-30} {4,-20} {5,-20}",
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("FullDomainName")),
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("IPAddress")),
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("ClientVersion")),
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("OSDescription")),
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("ComputerMake")),
                        sqldrReader.GetValue(sqldrReader.GetOrdinal("LastReportedStatusTime")));
```


0x03 探索 WSUS 攻击面

详细过程参考以下链接：
[us-15-Stone-WSUSpect-Compromising-Windows-Enterprise-Via-Windows-Update]

3.1、为什么要探索

- 更新通常可以通过非特权用户安装
- 可以作为提权途径
- 增加和降低 Windows 的攻击面

■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■

- 通过 WSUS 提供非微软的源代码

■■■■■■■
■■■■■■■■■■

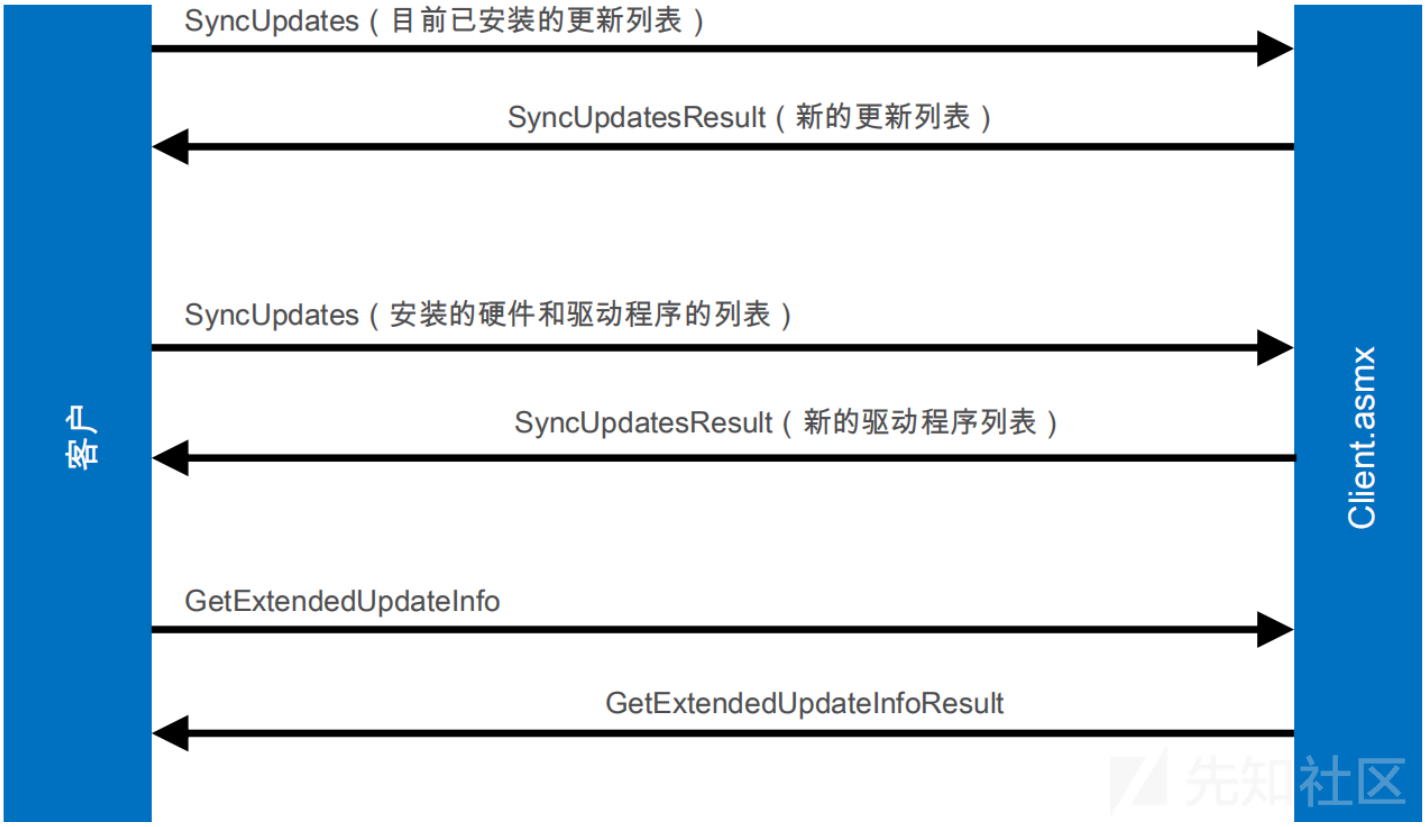
3.2、WSUS 安全

- 如果没有使用 SSL，则可以通过 MITM 更新流量（默认不启用 SSL，微软也不建议使用 SSL）
- 所有的更新都必须具备 Microsoft 签名。

那我估计大概或许可以这样做：

- 强制下载并安装驱动程序？
- 删除补丁方便攻击？
- 阻止更新？

3.3、WSUS SOAP 服务 - 检查更新



在这一过程，获取相关请求数据及扩展元数据，其实要想做 MITM，需要去了解一下这里的 SOAP协议，这个协议的内部交流是没有认证的。

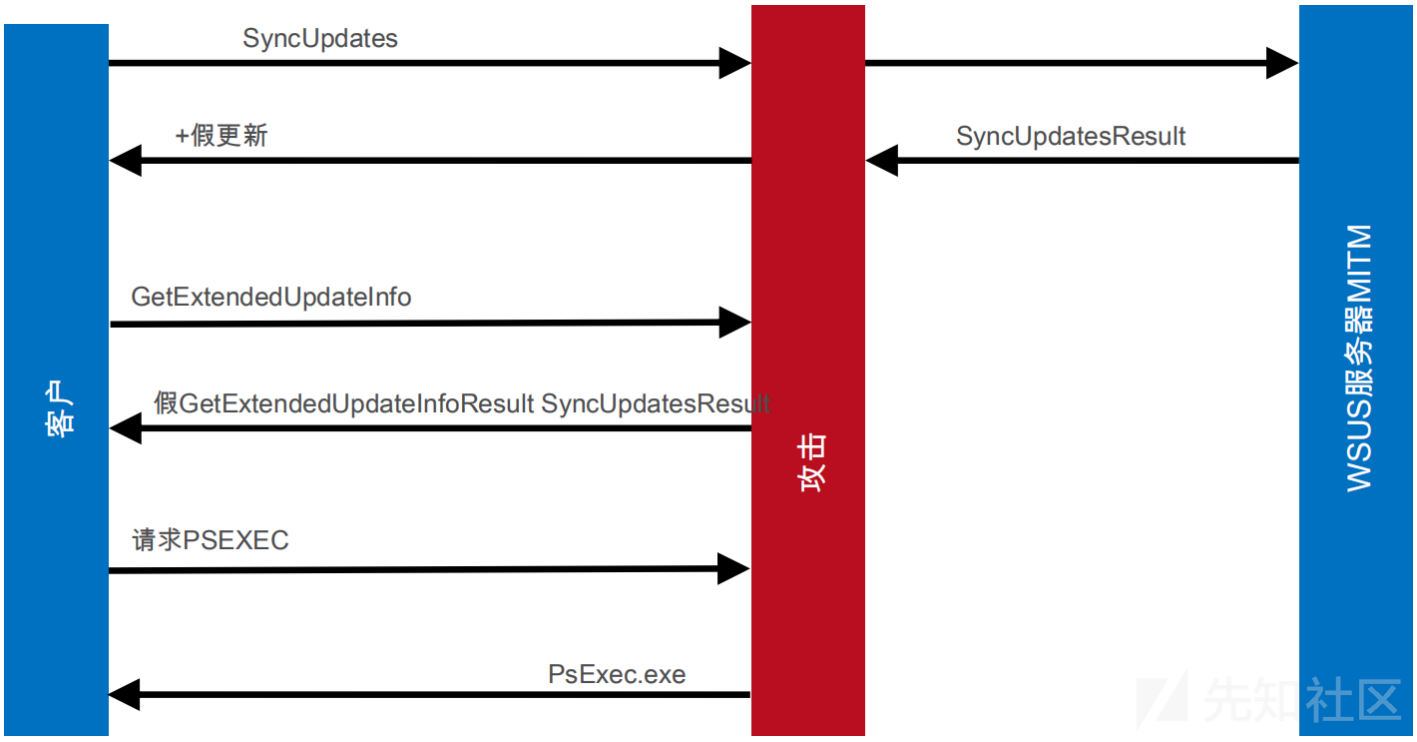
客户端与 WSUS 的认证是依靠 SSL，所以说没有使用 SSL的可以做 MITM。

0x04 WSUS 攻击向量 - CommandLineInstallation

在更新过程中，更新包

- 下载并运行任意带 Microsoft 签名的 exe
- 可提供任意命令行参数
- 安装时权限可为 system
- 带 Microsoft 签名的 Sysinternals 套装（PSEXEC）

以下是做 MITM 的示意图



过程也很简单，因为我们的程序充当了中间人，传话内容随意更改：

[redacted]

假设使用 psexec.exe，但是 psexec 本身的安全性也是个问题，现在多数 AV 程序都将其标记为 恶意软件。嘤嘤嘤。

```
PsExec.exe /accepteula cmd /c whoami > c:\whoami.txt
```

在元数据中则为

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

0x05 利用工具

- wsuspect-proxy
  WSUS 的安全问题是通过 Black Hat USA 2015 认识的，当时的 SUSpect – Compromising the Windows Enterprise via Windows Update 议题演示很是亮眼。Paul Stone（ @ pdjstone ）和来自 Context 的 Alex Chapman 明确指出，WSUS 默认使用的是 HTTP 协议，并没有SSL，所以任何人都可以对更新过程进行中间人攻击，以提供恶意的更新包。但是这个更新包必须是有 Microsoft 签名。但是这个也很好解决：SysInternals 的 PsExec 允许攻击者从已签名的 Windows 二进制文件执行任意 Windows 命令。
  工具由 Paul Stone 和 Alex Chapman 撰写，Github地址，具体操作可以看看其中的白皮书。
- WSUSpendu：https://github.com/AlsidOfficial/WSUSpendu
- Thunder_Woosus：https://github.com/ThunderGunExpress/Thunder_Woosus

0x06 WSUS 建议

- 启用 WSUS服务器上的 SSL
- 启用 客户端机器上的 SSL
- 独立网络，独立 WSUS

0x07 参考

Windows Server 2012 R2下补丁服务器部署与配置
us-15-Stone-WSUSpect-Compromising-Windows-Enterprise-Via-Windows-Update
WSUS pendu — Romain Coltel, Yves Le Provost
SSTIC2017-Article-wsus_pendu-coltel_le-provost

上一篇：angr 入门介绍（二） 下一篇：Linux内核漏洞利用：CVE-2...

1. 0 条回复
   - 动动手指，沙发就是你的了！

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板