Windows 10 IoT Core 远程命令执行漏洞验证及建议

安天微电子与嵌入式安全研发部

初稿完成时间：2019 年 03 月 07 日
首次发布时间：2019 年 03 月 25 日

©安天 版权所有，欢迎无损转载

目 录

©安天 版权所有，欢迎无损转载

1 概述

近日，安天微电子与嵌入式安全研发部（安天微嵌）针对 SafeBreach[1]公司披露的 Windows IoT[2]操作系统的安全漏洞进行了详细分析和验证。攻击者利用该漏洞可实现对目标设备的完全控制，如远程命令执行、文件上传/下载等。对此，安天微嵌成立了分析小组，分析验证了 SafeBreach 公司在 GitHub 中公布的该漏洞的原理及 POC，对该漏洞的影响范围进行了确认，并针对不同应用场景给出了相应的防护建议。

1.1 Windows IoT 系统简介

Windows 作为 IoT 市场中仅次于 Linux 系统的第二大系统，其 Windows IoT Core 是面向物联网领域的核心操作系统版本，覆盖了智能家居、智能医疗、智慧城市、智能物流等众多领域。Windows IoT 分为 IoT Core 和 IoT Enterprise 版本，而 Windows IoT Core 又分为 Stock Image 和 Custom Image 两个版本。其中，Stock Image 版本也被称为 Test Image 版本，其包含了用于开发及硬件兼容性测试用途的相关接口。

Windows IoT Core 系统目前支持的硬件平台包括：高通 DragonBoard 410c、树莓派 2、树莓派 3B、MinnowBoard Turbot、Aaeon Up Squared。

1.2 HLK 框架及 Sirep 协议简介

HLK（Hardware Lab Kit）是一个用于测试硬件设备及对应驱动程序和 Windows 系统之间兼容性的测试框架。PC 端软件 HLK Studio[3]中包含了该测试框架的 Server 部分，测试设备中包含了该测试框架的 Client 部分。用户可在 HLK Studio 中选择测试用例并发送到待测试的设备中进行测试。在底层实现中，HLK Studio 软件通过 Sirep 协议[4]与待测试设备进行交互。HLK Server 与 HLK Client 的关系如图 1-1 所示：

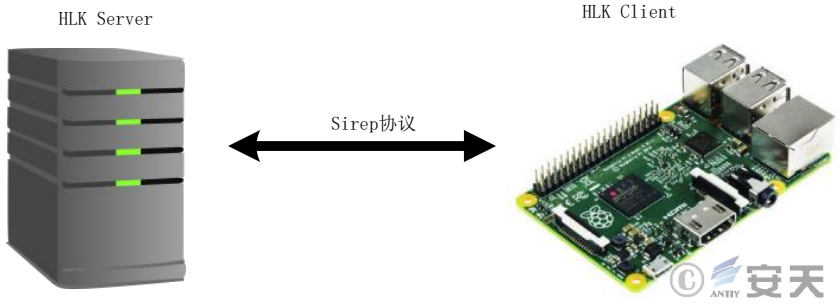

图 1-1 HLK Server 与 HLK Client 的关系

实现 Sirep 协议的 DLL 文件位于 Windows IoT Core 系统的 C:\..\testsirepsvc.dll 位置，该 DLL 实现了包括 HLK Studio 和 Windows IoT Core 系统间的通信功能，及执行 HLK Studio 下发至 Windows IoT Core 的测试任务等功能。而 Sirep 协议本身实现了如下功能：

获取 Windows IoT Core 系统信息；

执行 Windows IoT Core 系统命令；

下载 Windows IoT Core 系统中的文件；

上传文件到 Windows IoT Core 系统中；

获取 Windows IoT Core 系统中的文件属性信息。

2 影响范围

根据公开资料及分析小组实际验证，该漏洞目前主要影响 Windows IoT Core 的 Stock/Test Image[4]版本。若开发人员或厂商在最终发布的产品中使用了 Stock/Test Image 版本的系统，且该产品存在有线连接场景，则会受到此次披露的漏洞影响。构建 Custom 版本需要从 CA（Certificate Authority）购买签名证书，并使用该证书对 Custom 版本的系统进行签名，因为时间仓促，分析小组暂未对 Custom 版本的 Windows IoT 系统进行验证。

3 漏洞分析

本次分析过程使用树莓派 2B 和 Windows IoT Core（17763）版本作为验证环境。设备连接示意图如图 3-1 所示：

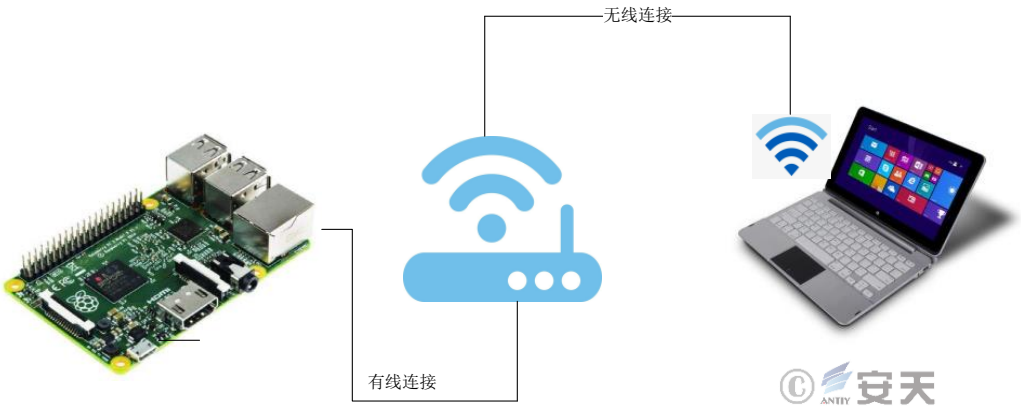

图 3-1 设备连接示意图

首先，可使用 Windows 10 IoT Core Dashboard[5]软件制作带有 Windows IoT Core 系统的 TF 卡，然后将 TF 卡插入到树莓派中，使用网线连接树莓派，通电后即可启动系统。系统启动后，Windows 10 IoT Core Dashboard 软件可自动发现局域网范围内的 Windows IoT 设备、执行启用远程 PowerShell、向设备中部署应用软件等。但这些操作需要首先通过 Windows IoT 系统管理员账号和密码认证才能正常使用。

在实现 Sirep 协议时，testsirepsvc.dll 的代码使用 TCP Socket 服务程序监听了 29817、29819、29820 端口，且在代码中并未对接收到的请求进行适当的权限检查，进而导致了在未经授权的情况下即可执行 Sirep 协议所实现的功能。

在 testsirepsvc.dll 中，实现对远程请求进行权限检查的函数名称为：

ControllerWSA::IsConnectionAllowed

在 Windows IoT Core 系统运行后，可通过 U 盘将该 DLL 文件拷贝到分析机中，使用 IDA 查看实现该函数的汇编代码，如图 3-2、图 3-3 所示：

```
t:10009354
t:10009354 ; int __fastcall ControllerWSA::IsConnectionAllowed(ControllerWSA *__hidden this, unsigned int)
t:10009354 _IsConnectionAllowed_ControllerWSA__QAAHI_Z
t:10009354                                         ; CODE XREF: SirepPipeServiceRoutine(void *,ulong)+36↓p
t:10009354                                         ; DATA XREF: .pdata:10012378↓o
t:10009354
t:10009354 var_24          = -0x24
t:10009354 var_1C          = -0x1C
t:10009354 var_18          = -0x18
t:10009354
t:10009354                 PUSH.W          {R4,R5,R11,LR}
t:10009358                 ADD.W           R11, SP, #8
t:1000935C                 BL              __security_push_cookie
t:10009360                 SUB             SP, SP, #0x1C
t:10009362                 MOV             R3, R1
t:10009364                 MOV             R5, R0
t:10009366                 MOV             R0, R3
t:10009368                 LDR             R3, =__imp_getsockname
t:1000936A                 MOVS            R2, #0x10
t:1000936C                 STR             R2, [SP,#0x24+var_24]
t:1000936E                 LDR             R3, [R3]
t:10009370                 MOV             R2, SP
t:10009372                 ADD             R1, SP, #0x24+var_1C
t:10009374                 MOVS            R4, #0
t:10009376                 BLX             R3
t:10009378                 CBZ             R0, loc_100093A0
t:1000937A                 LDR             R3, =__imp_WSAGetLastError
t:1000937C                 LDR             R3, [R3]
t:1000937E                 BLX             R3
```

©安天

图 3-2 ControllerWSA::IsConnectionAllowed 函数代码片段 1

```
t:1000939E                 B               loc_100093C2
t:100093A0 ; ---------------------------------------------------------------------------
t:100093A0
t:100093A0 loc_100093A0                            ; CODE XREF: ControllerWSA::IsConnectionAllowed(uint)+24↑j
t:100093A0                 ADD             R1, SP, #0x24+var_1C ; struct sockaddr_in *
t:100093A2                 ADD.W           R0, R5, #0x2C ; this
t:100093A6                 BL              _IsInterestingAddress_NetworkInterfaces__QAAHPAUsockaddr_in___Z ; Ne
t:100093AA                 CBNZ            R0, loc_100093C0
t:100093AC                 LDR             R3, =Microsoft_WindowsPhone_ToolConnectivity_SirepEnableBits
t:100093AE                 LDR             R3, [R3]
t:100093B0                 TST.W           R3, #0x400
t:100093B4                 BEQ             loc_100093C2
t:100093B6                 LDR             R1, =SirepServer_WarningRejectedIncomingInterface
t:100093B8                 LDR             R2, [SP,#0x24+var_18]
t:100093BA                 BL              McTemplateU0d
t:100093BE                 B               loc_100093C2
t:100093C0 ; ---------------------------------------------------------------------------
```

©安天

图 3-3 ControllerWSA::IsConnectionAllowed 函数代码片段 2

以上代码片段解释了 testsirepsvc.dll 如何对接收到的请求进行权限检查的逻辑，程序逻辑仅仅判断 getsockname 函数返回的 SOCKADDR_IN 数据结构是否为有线网卡的 IP 地址，也就是说 testsirepsvc.dll 认为所有来自有线网络的请求都是合法的请求，这个过程并不需要用户输入用户名和密码进行认证。

在 testsirepsvc.dll 中的服务程序接收到 HLK Studio 发送的命令数据后，会经过 SirepPipeServiceRoutine 函数进行分流，在该函数中通过命令类型字段将不同的命令分流到不同的函数中进行执行，不同的命令类型对应的执行函数名称分别为：

SirepGetSystemInformationFromDevice;

SirepPutFileOnDevice;

SirepGetFileFromDevice;

SirepGetFileInformationFromDevice;

SirepLaunchWithOutput

实现该命令分流过程的 SirepPipeServiceRoutine 函数汇编代码如图 3-4 所示：

©安天 版权所有，欢迎无损转载

```
loc_1000D35C                            ; CODE XREF: SirepPipeServiceRoutine(
                MOVS            R2, #8
                ADD             R1, SP, #0x4C+var_44
                MOV             R0, R5
                STR.W           R8, [SP,#0x4C+var_4C]
                BL              SirepProtocol2ReceivePacketWithTimeout
                MOV             R4, R0
                CMP             R4, #0
                BLT             loc_1000D412
                LDR             R3, [SP,#0x4C+var_44]
                CMP             R3, #0x1E
                BGT             loc_1000D3B4
                BEQ             loc_1000D3AA
                CMP             R3, #0xA
                BEQ             loc_1000D3A0
                CMP             R3, #0x14
                BEQ             loc_1000D39A
                CMP             R3, #0x17
                BEQ             loc_1000D396
                CMP             R3, #0x18
                BNE             loc_1000D406
                MOVS            R3, #1  ; int
```

©安天

图 3-4 SirepPipeServiceRoutine 函数命令分流代码

4 POC 执行验证

分析小组通过树莓派 2B 和 Windows IoT Core（17763）（Stock/Test Image）搭建了验证环境，结合上述的分析过程对公开的 POC[6]进行了验证。

上传文件 POC 执行示例如图 4-1、图 4-2 所示：

```
PS D:\SirepRAT> python SirepRAT.py 192.168.0.100 Pu
tFileOnDevice --remote_path "C:\Windows\System32\up
loaded.txt" --data "Hello Windows IoT!"
<HResultResult | type: 1, payload length: 4, HResul
t: 0x0>
<HResultResult | type: 1, payload length: 4, HResul
t: 0x0>
```

©安天

图 4-1 上传文件 POC 执行示例

如图 4-1 所示：该 POC 在 C:\Windows\System32\目录下创建了一个名为 uploaded.txt 的文件，文件内容为“Hello Windows IoT!”。

```
[192.168.0.100]: PS C:\Windows\System32>
[192.168.0.100]: PS C:\Windows\System32> cat .\uploaded.txt
H e l l o   W i n d o w s   I o T !
```

©安天

图 4-2 查看上传文件结果

如图 4-2 所示，在执行图 4-1 所示的 POC 命令后，使用 cat 命令可查看 C:\Windows\System32\目录下的文件名为 uploaded.txt 的文件内容。

远程执行系统命令 POC 执行示例如图 4-3 所示：

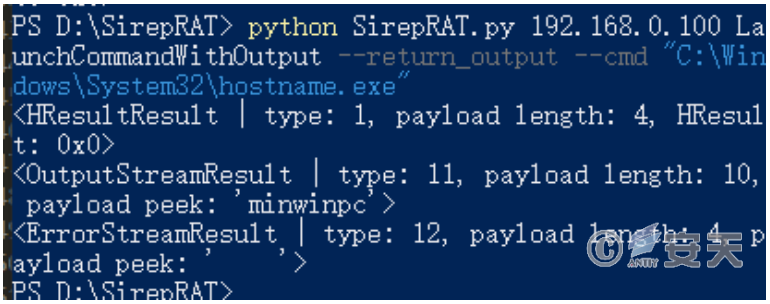

图 4-3 远程执行系统命令 POC 执行示例

如图 4-3 所示，该 POC 实现了远程命令执行，在目标设备中执行了 hostname 命令，并返回了命令执行的结果，即“minwinpc”。

经过验证，SafeBreach 公司安全研究人员公开的 POC 能够实现对 Stock/Test Image 版本的 Windows IoT Core 系统的上传文件和执行系统命令，Stock/Test Image 版本的 Windows IoT Core 系统存在严重的安全隐患。

5 防护建议

本次验证的漏洞能够不经授权就可以在受影响系统设备上执行上传文件和执行系统命令等高危动作，恶意软件通过本漏洞的利用很容易劫持设备成为僵尸网络的一员，成为黑客发起网络攻击的武器之一；设备也能够被黑客控制成为挖矿中的一部分；同时由于 IoT 设备应用于各行各业，一旦受到控制最直接的就是影响设备的正常运行，进而对生产生活造成影响；同时黑客也可以通过设备作为跳板进一步入侵 IoT 设备所在网络进行病毒传播、情报窃取和网络破坏等危险行为，对目标网络造成严重威胁。

虽然本次验证的漏洞仅适用于 Stock/Test Image 版本的 Windows IoT Core 系统，但由于构建 Custom 版本需要从 Certificate Authority (CA)购买签名证书，并使用该证书对 Custom 版本的系统进行签名，厂商可能出于成本或其它方面考虑直接使用 Stock/Test Image 版本的 Windows IoT Core 系统进行产品发布，也就是说以 Stock/Test Image 版本的 Windows IoT Core 系统 IoT 设备可能已经广泛进入供应链。并且 IoT 设备在现实应用场景中进行固件升级较为困难，容易被忽视。

©安天 版权所有，欢迎无损转载

综合上述情况为了有效降低漏洞所带来的威胁，提高产品安全性的同时，有效提升产品所在网络的安全防护能力，保障客户价值。我们结合漏洞的分析和验证情况给出三点安全建议，具体如下：

建议一： 产品实际的上线过程应该严格按照官方要求的研发、测试和发布流程规范操作，使用 Custom Image 而非 Stock/Test Image 版本的 Windows IoT 系统作为实际产品的发布系统，能够有效避免本次或其它未被发现的 Stock/Test Image 版本系统漏洞所产生的影响。

建议二：本漏洞所涉及到的服务使用 29817、29819、29820 三个端口，且涉及到的服务仅用于研发阶段的兼容性测试，并不是实际产品所使用的功能。在暂时无法升级固件的情况下，并确保实际产品中没有依赖相应端口的功能，以防止在关闭相应端口后影响设备的正常使用，则可以临时在 Windows IoT Core 系统防火墙中将兼容性测试服务所使用的 29817、29819、29820 三个端口进行阻断，也可暂时避免本次披露漏洞所产生的影响。但仍需要尽快升级固件修补漏洞，才能有效避免本次或其它未被发现的 Stock/Test Image 版本系统漏洞所产生的影响。在 Windows IoT Core 系统临时阻断端口的命令如下：

```
netsh advfirewall firewall set rule name=all localport=29817 protocol=tcp new enable=no
netsh advfirewall firewall set rule name=all localport=29819 protocol=tcp new enable=no
netsh advfirewall firewall set rule name=all localport=29820 protocol=tcp new enable=no
```

建议三： 根据 IoT 设备所实现功能的技术特征并结合实际运行环境，详细梳理可以访问设备的 IP 列表、端口列表、访问协议类型，以及设备可以向外主动连接的协议类型、IP 列表和端口列表，结合梳理结果使用边界防火墙产品或设备专用防火墙产品配置相应的双向 IP 地址、端口和协议的白名单访问规则列表，可最大限度的保障 IoT 设备的访问安全。该方法虽然能够有效保障 IoT 设备的访问安全，降低漏洞被利用的可能，但并未根除漏洞风险，所以尽快升级固件修补漏洞，才能有效避免本次或其它未被发现的 Stock/Test Image 版本系统漏洞所产生的影响。

©安天 版权所有，欢迎无损转载

附录一：参考链接

[1] SafeBreach

https://safebreach.com/News-Post/SafeBreach-Labs-Discloses-New-Microsoft-Windows-IoT-Core-Weakness-and-Exploit

[2] Windows IoT

https://docs.microsoft.com/en-us/windows/iot-core/windows-iot-core

[3] HLK Studio

https://docs.microsoft.com/en-us/windows-hardware/test/hlk/user/hlk-studio

[4] Sirep 协议和 Stock Image

https://github.com/SafeBreach-Labs/SirepRAT/blob/master/docs/SirepRAT_RCE_as_SYSTEM_on_Windows_IoT_Core_White_Paper.pdf

[5] Windows 10 IoT Core Dashboard

https://docs.microsoft.com/en-us/windows/iot-core/connect-your-device/iotdashboard

[6]POC

https://github.com/SafeBreach-Labs/SirepRAT

©安天 版权所有，欢迎无损转载

附录二：关于安天

安天是引领威胁检测与防御能力发展的网络安全国家队，始终坚持自主先进能力导向，依托下一代威胁检测引擎等先进技术和赛博超脑大平台工程能力积累，研发了智甲、镇关、探海、捕风、追影、拓痕等系列产品，为客户构建端点防护、边界防护、流量监测、导流捕获、深度分析、应急处置的安全基石。安天致力于为客户建设实战化的态势感知体系，依托全面持续监测能力，建立系统与人员协同作业机制，指挥网内各种防御机制联合响应威胁，实现从基础结构安全、纵深防御、态势感知与积极防御到威胁情报的有机结合，协助用户开展深度结合与全面覆盖的体系化网络安全规划与建设，支撑起协同联动的实战化运行，赋能用户筑起可对抗高级威胁的网络安全防线。

安天为网信主管部门、军队、保密、部委行业和关键信息基础设施等高安全需求客户，提供整体安全解决方案，产品与服务为载人航天、探月工程、空间站对接、大飞机首飞、主力舰护航、南极科考等提供了安全保障。参与了 2005 年后历次国家重大政治社会活动的安保工作，并多次获得杰出贡献奖、安保先进集体等称号。

安天是全球基础安全供应链的核心赋能方，全球近百家著名安全企业、IT 企业选择安天作为检测能力合作伙伴，安天的威胁检测引擎为全球超过三十万台网络设备和网络安全设备、超过十五亿部智能终端设备提供了安全检测能力。其中，安天的移动检测引擎是第一个获得权威国际评测奖项的中国产品。

安天是中国应急响应体系中重要的企业节点，在“红色代码”、“口令蠕虫”、“心脏出血”、“破壳”、“魔窟”等重大安全威胁和病毒疫情方面，实现了先发预警和全面应急响应。安天针对“方程式”、“白象”、“海莲花”、“绿斑”等几十个高级网空威胁行为体及其攻击行动，进行持续监测和深度解析，协助客户在“敌情想定”下形成有效防护，通过深度分析高级网空威胁行为体的作业能力，安天建立了以实战化对抗场景为导向的能力体系。

安天实验室更多信息请访问： http://www.antiy.com（中文）

http://www.antiy.net （英文）

安天企业安全公司更多信息请访问： http://www.antiy.cn

安天移动安全公司（AVL TEAM）更多信息请访问： http://www.avlsec.com

©安天 版权所有，欢迎无损转载