IBM Security

Into The Core

IN-DEPTH EXPLORATION OF WINDOWS 10 IOT CORE

Paul Sabanal
X-Force Advanced Research
IBM Security

August 3, 2016

IBM

Introduction

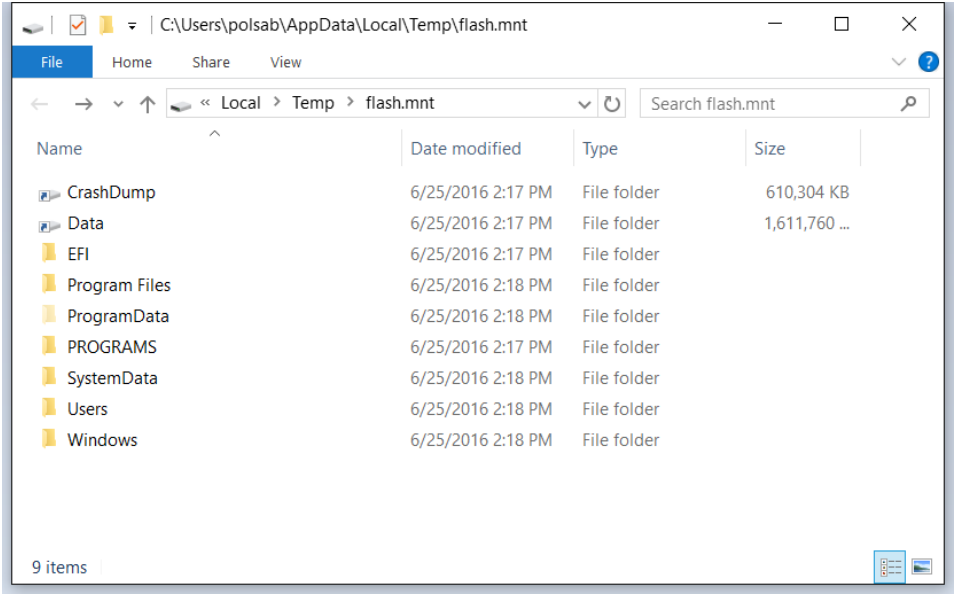
Overview

Raspberry Pi 2 & 3	• ARM • 32-bit •On-board Wi-fi and Bluetooth, Ethernet • 4 x USB 2.0
Minnowboard Max	• x86 • 32-bit • Ethernet • 1 x USB 2.0, 1 x USB 3.0
Dragonboard 410c	• ARM • On-board Wi-fi and Bluetooth • 2 x USB 2.0

Internals

Internals > FFU

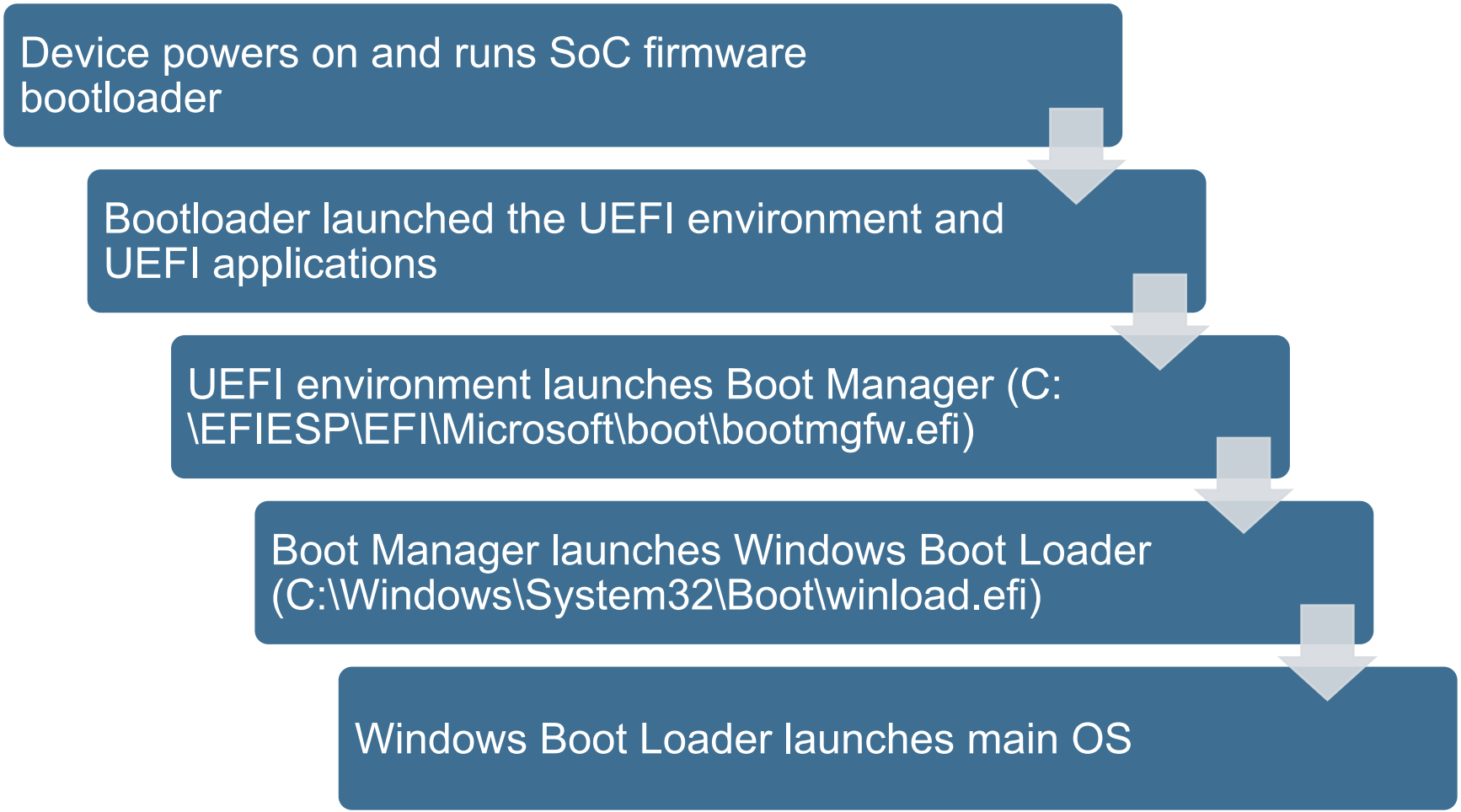
```
C:\>ImgMount.exe "c:\Program Files (x86)\Microsoft IoT\FFU\MinnowBoardMax\flash.ffu"

WP8 ROM Image Tools v.1.0.204
htc ROM Image Editor (⌐) 2007-2012 AnDim & XDA-Developers
ImgMount Tool v.1.0.15

(htcRIE) Mounting the image file : 'c:\Program Files (x86)\Microsoft IoT\FFU\MinnowBoardMax\flash.ffu'
Loading .FFU image ... ok
Creating virtual disk ... ok
Mounting MainOS partition as : '\\flash.mnt\' ... ok
(htcRIE) Successfully mounted an image file.
```



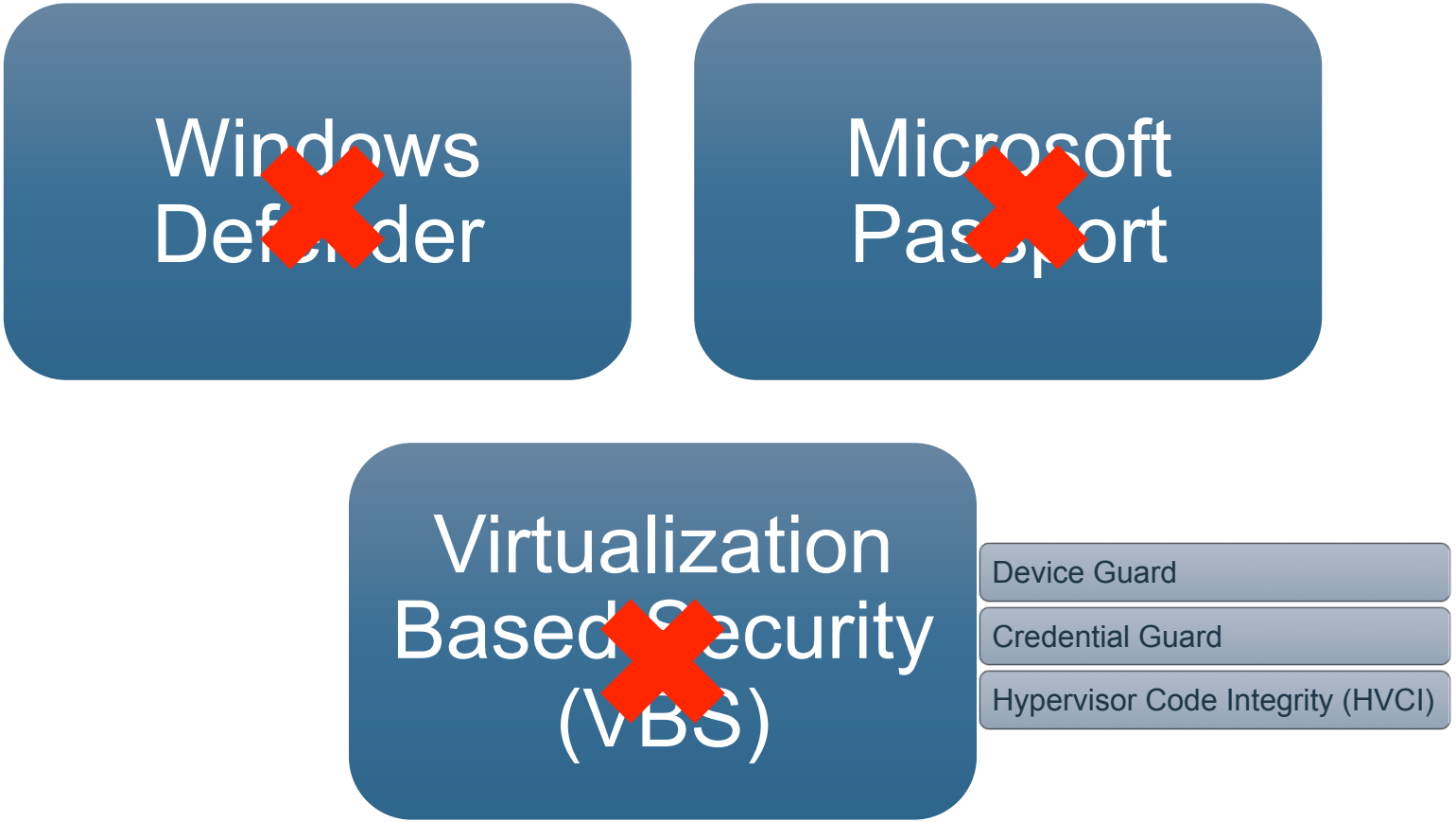
Internals > Partition Layout

Partition	File System	Mount Point	Contents
EFI System Partition	FAT	C:\EFIESP	Boot manager, boot configurations, UEFI applications
Crash dump partition	NTFS	D:	Crash dump data
Main OS	NTFS	C:	OS, registry hives, OEM applications
Data partition	NTFS	U:	Applications, application data,user data

Internals > Boot Process



Internals > Security



Internals > Security

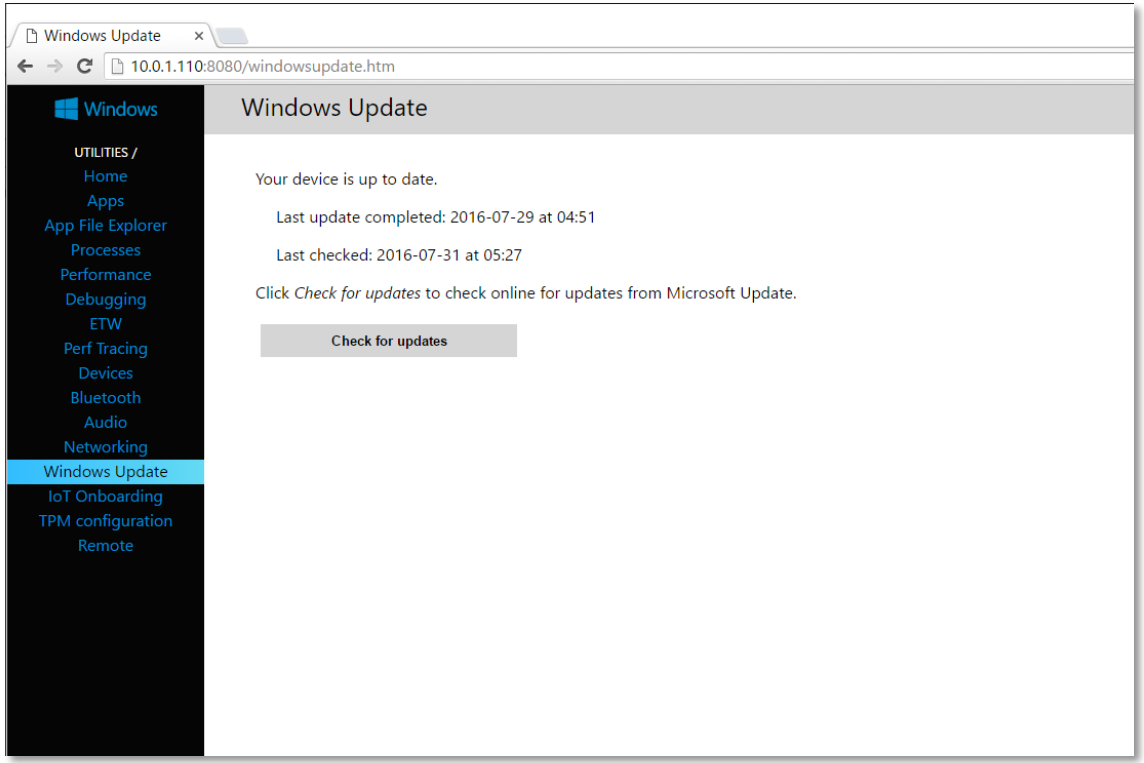
Type	Description
Firmware TPM	TPM implemented in the SoC
Discrete TPM	Chip module that can be attached to a board
Software TPM	Software emulated TPM used in development

- Secure Boot
  - Prevents device tampering during boot
  - Stops the system from running unverified binaries
  - Protects against boot kits, rootkits, and other low level malware

- BitLocker
  - Lightweight version of BitLocker
  - Encryption of user and system files

Internals > Windows Update

- Automatic forced update
- Check for updates through “Windows Update” tab of Windows Device Portal
- Pro edition allows deferred updates

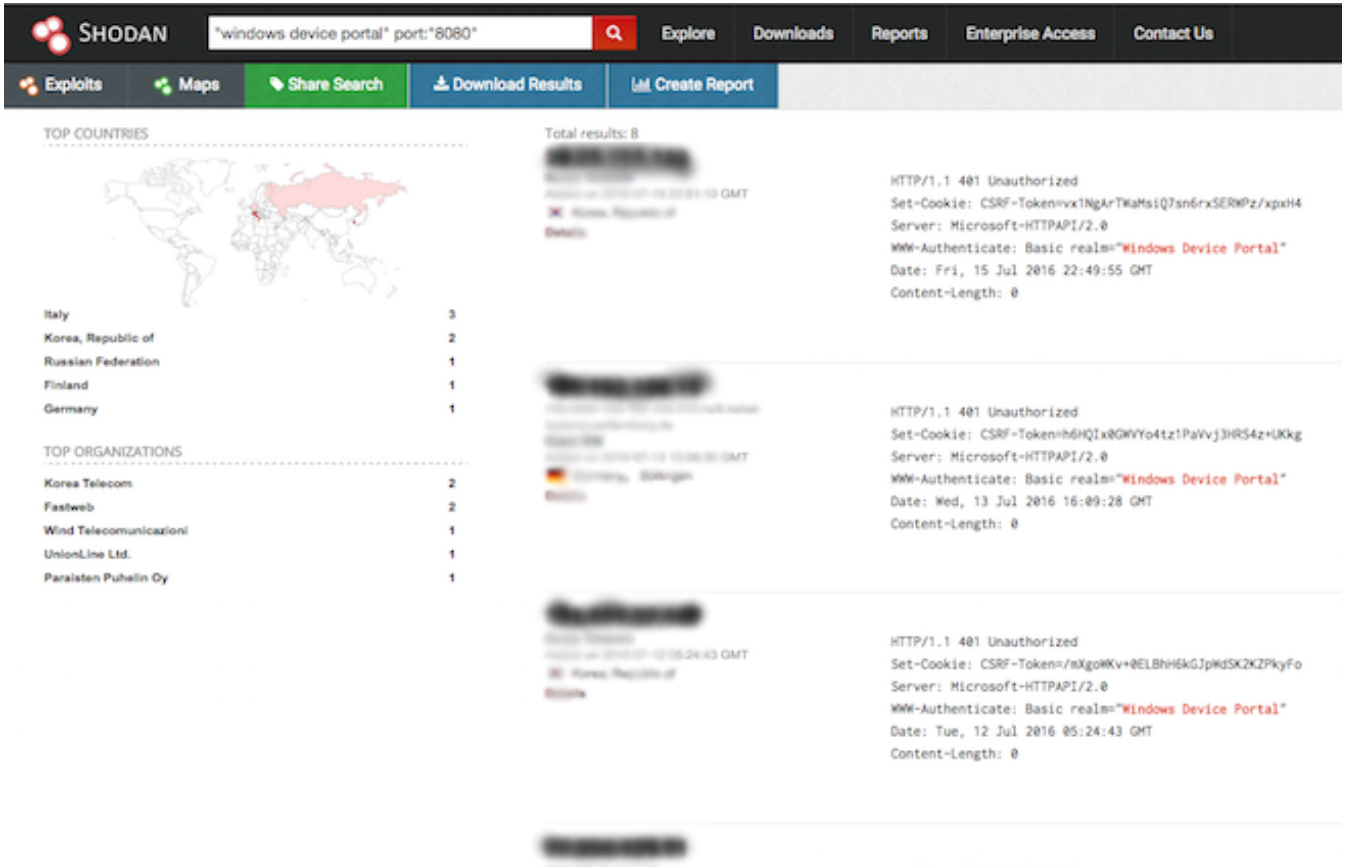


Attack Surface

Attack Surface > Network services

```
Starting Nmap 7.12 ( https://nmap.org ) at 2016-07-13 01:33 Malay
Peninsula Standard Time
Nmap scan report for 10.0.1.108
Host is up (0.020s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
8080/tcp open  http-proxy
MAC Address: B8:27:EB:B5:A9:E0 (Raspberry Pi Foundation)

Nmap done: 1 IP address (1 host up) scanned in 3.24 seconds
```

Attack Surface > Network services



Attack Surface > Network services

- SSH
  - Enabled by default
- Windows File Sharing
  - Enabled by default
- Windows IoT Remote Server
  - Remote UI client installed from Windows Store
  - Can be enabled through the Remote tab in the Windows Device Portal
  - No authentication

Attack Surface > Windows Device Portal

Utility	Function
Home	Device information, change device name/password, timezone settings
Apps	Install/uninstall of apps
App File Explorer	File explorer for installed apps locations
Processes	Running processes list, process memory usage, and process termination
Performance	Real time graphical display of CPU and I/O usage
Debugging	Starting VS remote debugger, downloading of live kernel and process dumps
ETW	Event tracing
Perf Tracing	Trace logging of CPU, disk, and memory usage
Devices	Device manager for peripherals attached to the device
Bluetooth	Bluetooth device search
Audio	Device speaker and microphone volume adjustments
Networking	WiFi configuration
Windows Update	Last update timestamp, check for updates
IoT Onboarding	Internet Connection Sharing settings, SoftAP settings, AllJoyn onboarding settings
TPM Configuration	TPM installation, configuration, and provisioning
Remote	Enable Windows IoT Remote Server

Attack Surface > Device drivers

- Drivers for built-in or external peripherals

- Drivers for wireless adapters
  - Wifi
  - Bluetooth
  - ZigBee
  - Z-Wave

- Successful exploitation often results in kernel level privilege

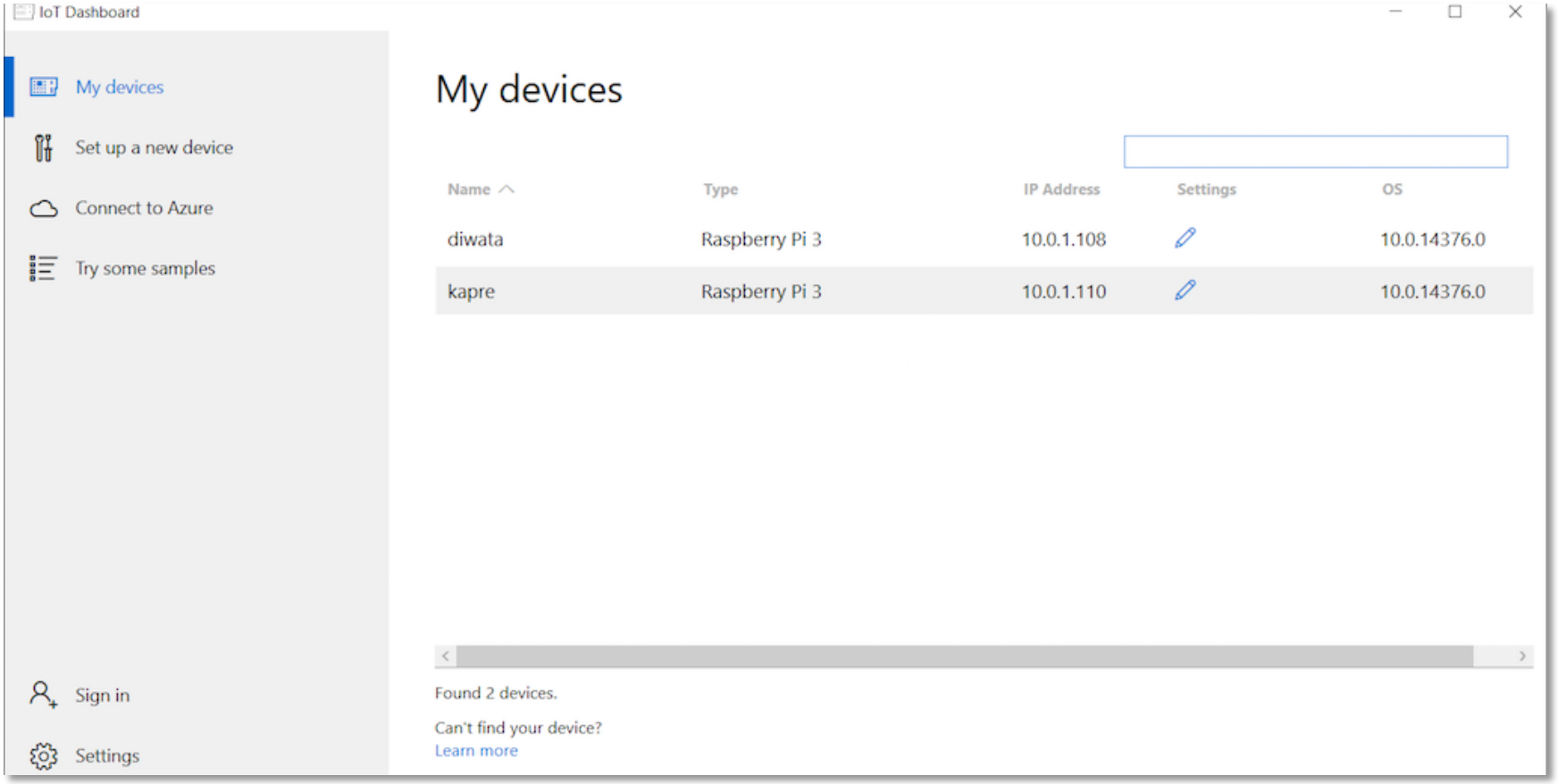
Attack Surface > Malware

- Password guessing/brute forcing of login credentials
- Lateral infection coming from other machines

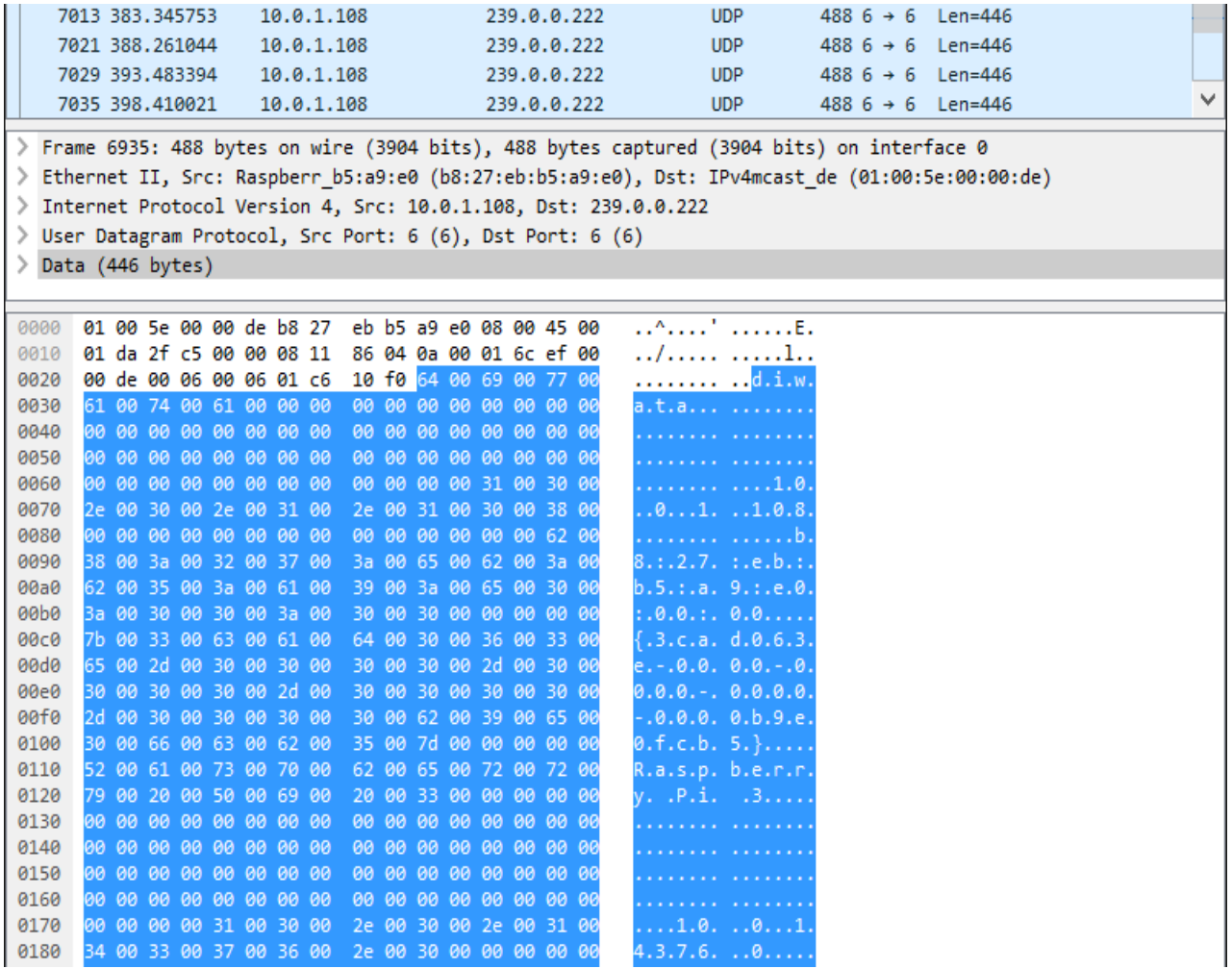
```
C:\>mimikatz.exe

  .#####.   mimikatz 2.1 (x64) built on Jul 11 2016 00:32:57
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 20 modules * * */

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::ssp

Authentication Id : 0 ; 247557 (00000000:0003c705)
Session           : Interactive from 1
User Name         : polsab
Domain            : DESKTOP-39HUL88
Logon Server      : (null)
Logon Time        : 7/20/2016 6:15:59 PM
SID               : S-1-5-21-4294890806-594742593-2658599142-1001
        ssp :
         [00000000]
         * Username : Administrator
         * Domain   : 10.0.1.108
         * Password : diwata
```

Hacking

Hacking > Device Discovery



Hacking > Device Discovery



Offset	Description
0	Device name
0x42	IP address
0x64	MAC address
0x96	Board serial number
0xe6	Device Type
0x14a	OS version
0x1ae	Device architecture

Hacking > PowerShell

- Remote device administration and configuration

- Built-in and 3rd party tools for penetration testing and reversing. Ex:
  - CimSweep
  - Autoruns

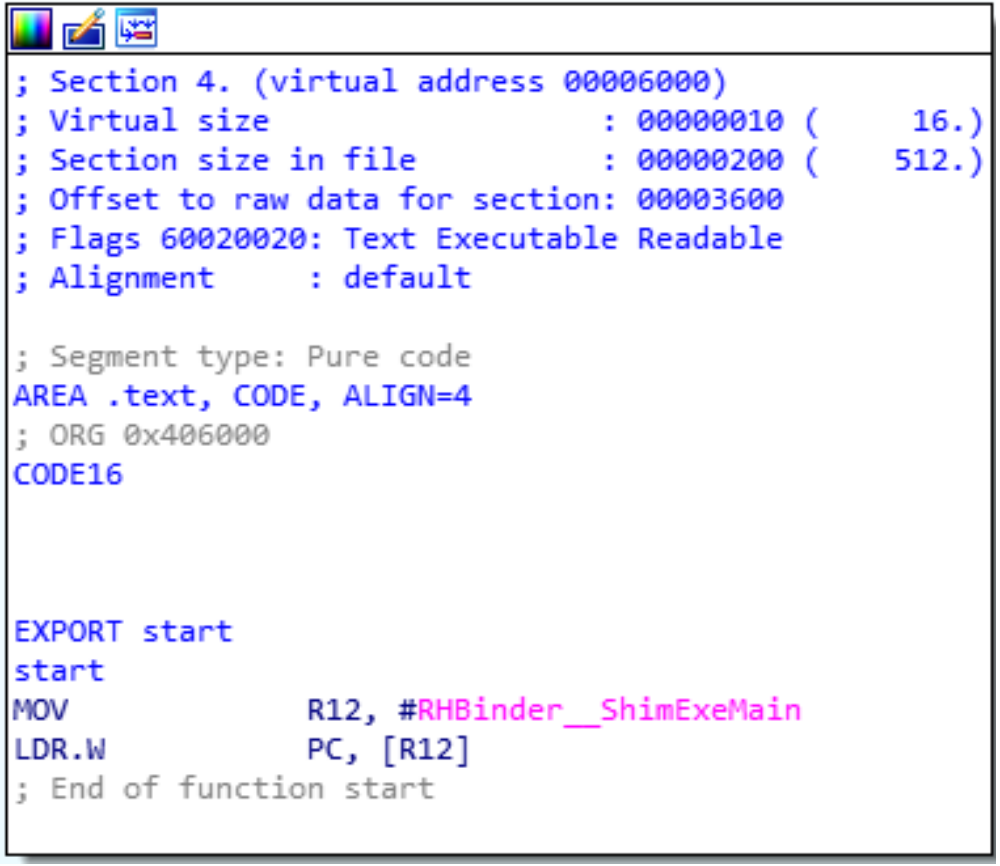
```
PS C:\WINDOWS\system32> $CimSessionPi2 = New-CimSession -ComputerName
10.0.1.110 -Credential Administrator
PS C:\WINDOWS\system32> Get-CSRegistryAutoStart -CimSession $CimSessionPi2

Path            : HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
AutoRunEntry    : Shell
ImagePath       : IotShell.exe
Category        : Logon
PSComputerName  : 10.0.1.110

Path            : HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
AutoRunEntry    : Userinit
ImagePath       : userinit.exe
Category        : Logon
PSComputerName  : 10.0.1.110

Path            : HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
AutoRunEntry    : VMApplet
ImagePath       : SystemPropertiesPerformance.exe /pagefile
Category        : Logon
PSComputerName  : 10.0.1.110

Path            : HKLM\SYSTEM\CurrentControlSet\Control\Session Manager
AutoRunEntry    : BootExecute
ImagePath       : autocheck autochk *
Category        : BootExecute
PSComputerName  : 10.0.1.110
<snip>
```

Hacking > Static analysis

- UWP apps can be found in Data partition (U:\, also linked with C:\Data)
- App installed in Programs \WindowsApps
- Lib DLLs and XBF (binary XAML)
- Assets folder
  - Images
  - Fonts
  - etc

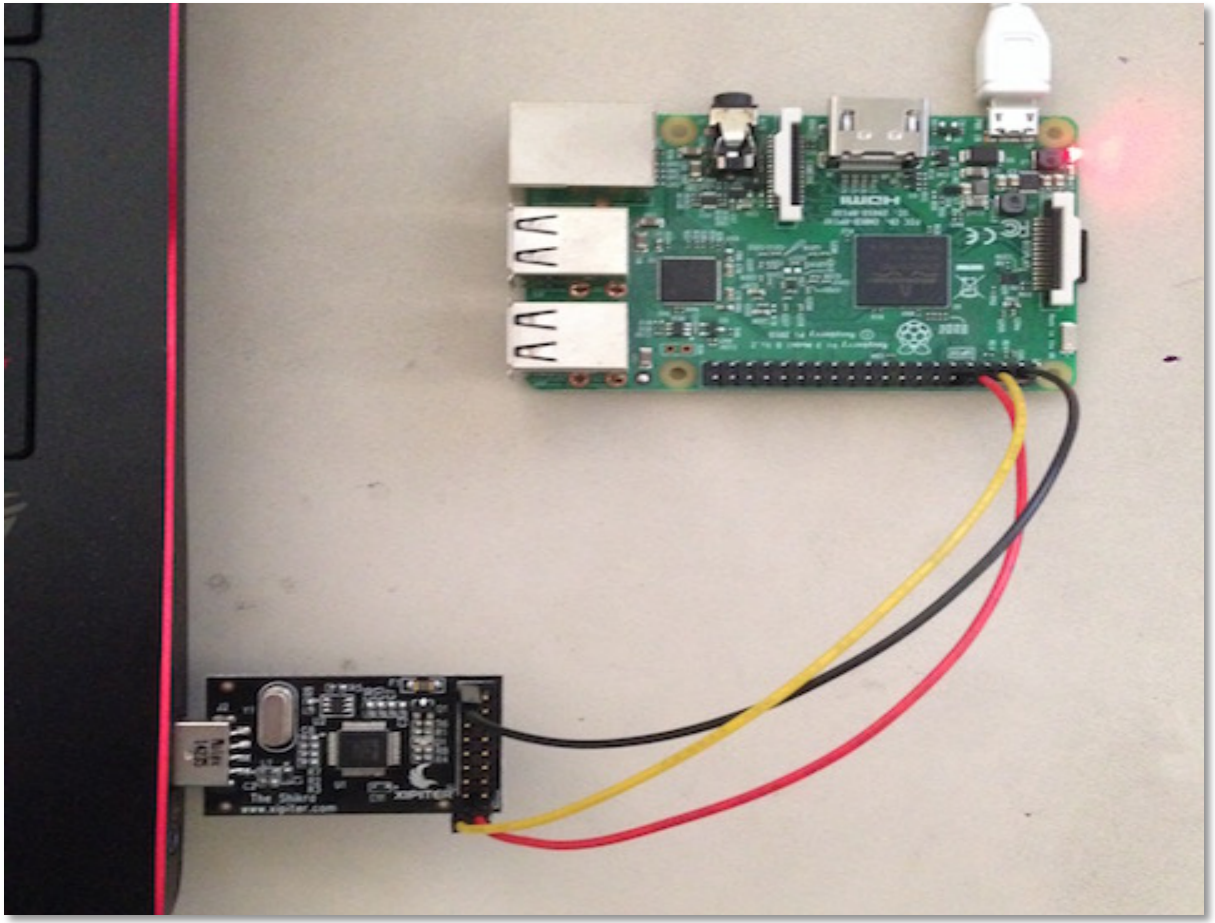
Filename	Description
<app_name>.exe	App startup stub
<app_name>.dll	App code
AppManifest.xml	UWP app package manifest
AppBlockMap.xml	Cryptographic block hashes for files in package
AppxSignature.p7x	App package digital signature file

Hacking > Static analysis



```
; Section 4. (virtual address 00006000)
; Virtual size                   : 00000010 (     16.)
; Section size in file           : 00000200 (    512.)
; Offset to raw data for section: 00003600
; Flags 60020020: Text Executable Readable
; Alignment     : default

; Segment type: Pure code
AREA .text, CODE, ALIGN=4
; ORG 0x406000
CODE16



EXPORT start
start
MOV             R12, #RHBinder__ShimExeMain
LDR.W           PC, [R12]
; End of function start
```

Hacking > Kernel debugging



Hacking > Kernel debugging

```
# Enable serial debugging
bcdedit -dbgsettings serial
# Turn on debugging
bcdedit -debug on
```

```
Get-WMIObject Win32_pnpentity | ? Name -like "*Serial*COM*"
```

```
__GENUS                       : 2
__CLASS                       : Win32_PnPEntity
__SUPERCLASS                  : CIM_LogicalDevice
__DYNASTY                     : CIM_ManagedSystemElement
__RELPATH                     : Win32_PnPEntity.DeviceID="FTDIBUS\\VID_0403+PID_6014+5&3278CBC5&0&3\\0000"
__PROPERTY_COUNT              : 26
__DERIVATION                  : {CIM_LogicalDevice, CIM_LogicalElement, CIM_ManagedSystemElement}
__SERVER                      : DESKTOP-39HUL88
__NAMESPACE                   : root\cimv2
__PATH                        : \\DESKTOP-39HUL88\root\cimv2:Win32_PnPEntity.DeviceID="FTDIBUS\
\VID_0403+PID_6014+5&3278CBC5&0&3\\0000"
Availability                  :
Caption                       : USB Serial Port (COM3)
ClassGuid                     : {4d36e978-e325-11ce-bfc1-08002be10318}
CompatibleID                  :
```

Hacking > Kernel debugging

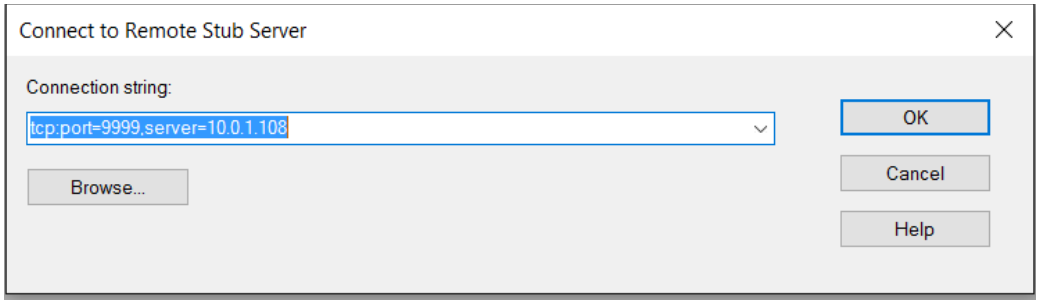
```
# PORT is the COM port number used by your USB-to-serial adapter
windbg.exe -k com:port=<PORT>,baud=921600
```

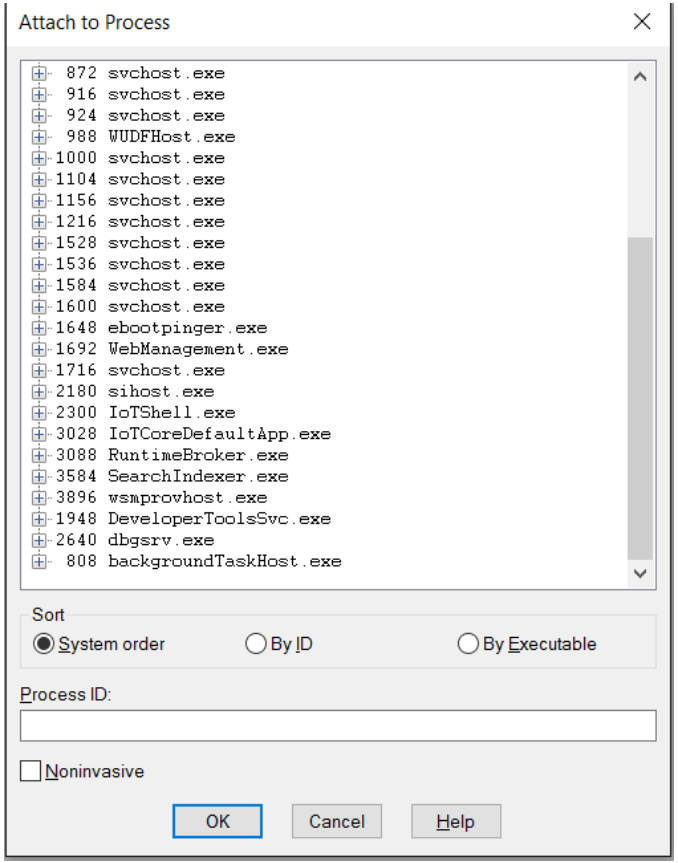
```
Microsoft (R) Windows Debugger Version 10.0.10586.567 X86
Copyright (c) Microsoft Corporation. All rights reserved.
Opened \\.\com3
Waiting to reconnect...
```

```
Connected to Windows 10 14393 ARM (NT) Thumb-2 target at (Sun Jul 24 19:32:43.111 2016 (UTC + 8:00)), ptr64 FALSE
Kernel Debugger connection established.
Symbol search path is: srv*
Executable search path is:
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlmp.exe -

Windows 10 Kernel Version 14393 MP (1 procs) Free ARM (NT) Thumb-2
Built by: 14393.0.armfre.rs1_release.160715-1616
Machine Name:
Kernel base = 0x80c1b000 PsLoadedModuleList = 0x80e07c78
System Uptime: 0 days 0:00:00.000
Break instruction exception - code 80000003 (first chance)
*******************************************************************************
*                                                                             *
*   You are seeing this message because you pressed either                    *
*       CTRL+C (if you run console kernel debugger) or,                       *
*       CTRL+BREAK (if you run GUI kernel debugger),                          *
*   on your debugger machine's keyboard.                                      *
*                                                                             *
*                   THIS IS NOT A BUG OR A SYSTEM CRASH                       *
*                                                                             *
* If you did not intend to break into the debugger, press the "g" key, then   *
* press the "Enter" key now.  This message might immediately reappear.  If it *
* does, press "g" and "Enter" again.                                          *
*                                                                             *
*******************************************************************************
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlmp.exe -
nt!DbgBreakPointWithStatus:
80c40d90 defe      __debugbreak
```

Hacking > User mode debugging

```
# PORT is the local port you want dbgsrv to listen on
C:\Windows\System32\Debuggers\dbgsrv.exe -t tcp:port=<PORT>
```

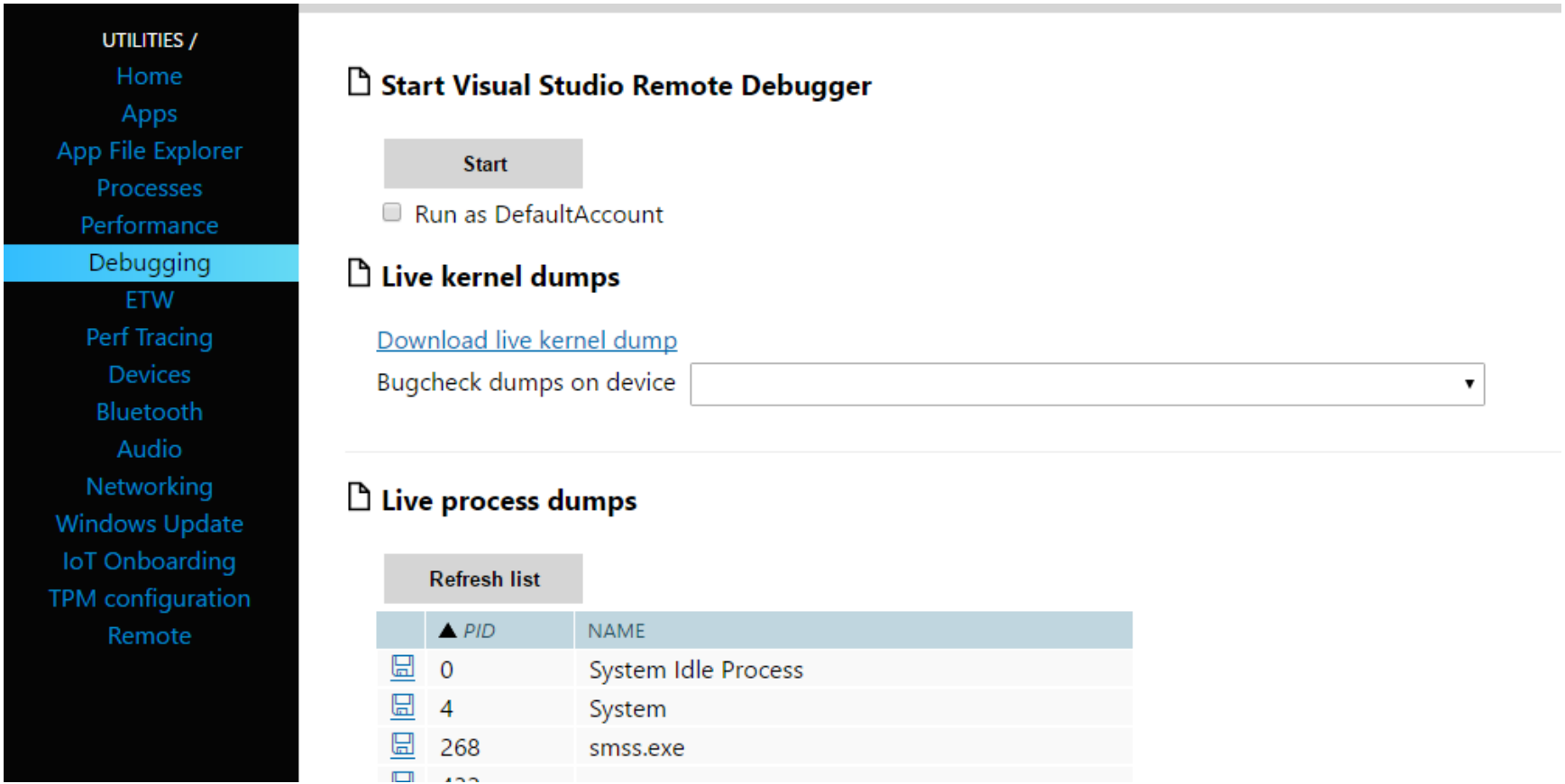




Hacking > User mode debugging

```
Microsoft (R) Windows Debugger Version 10.0.10586.567 X86
Copyright (c) Microsoft Corporation. All rights reserved.

*** wait with pending attach
Symbol search path is: srv*
Executable search path is:
ModLoad: 01110000 011db000   C:\windows\system32\WebManagement.exe
ModLoad: 77400000 77565000   C:\windows\SYSTEM32\ntdll.dll
ModLoad: 77270000 773fe000   C:\windows\System32\KERNELBASE.dll
<snip...>
(69c.280): Break instruction exception - code 80000003 (first chance)
ntdll!DbgBreakPoint:
77422740 defe      __debugbreak
0:005> !peb
PEB at 00928000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         01110000
    Ldr                       774eb9e0
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00c41738 . 00c4fcd0
    Ldr.InLoadOrderModuleList:           00c41810 . 00c4fcc0
    Ldr.InMemoryOrderModuleList:         00c41818 . 00c4fcc8
            Base TimeStamp                     Module
         1110000 57898ebe Jul 16 09:32:46 2016 C:\windows\system32\WebManagement.exe
        77400000 57898ba5 Jul 16 09:19:33 2016 C:\windows\SYSTEM32\ntdll.dll
        77270000 57898c4c Jul 16 09:22:20 2016 C:\windows\System32\KERNELBASE.dll
<snip...>
0:005> u $exentry
WebManagement+0xa6631:
011b6630 e92d4800 push        {r11,lr}
011b6634 46eb     mov         r11,sp
011b6636 f000fb65 bl          WebManagement+0xa6d04 (011b6d04)
011b663a e8bd4800 pop         {r11,lr}
011b663e f7ffbf25 b.w         WebManagement+0xa648c (011b648c)
011b6642 0000     movs        r0,r0
011b6644 f24c6c64 mov         r12,#0xC664
011b6648 f2c01c1c movt        r12,#0x11C
```

Hacking > Crash Dump

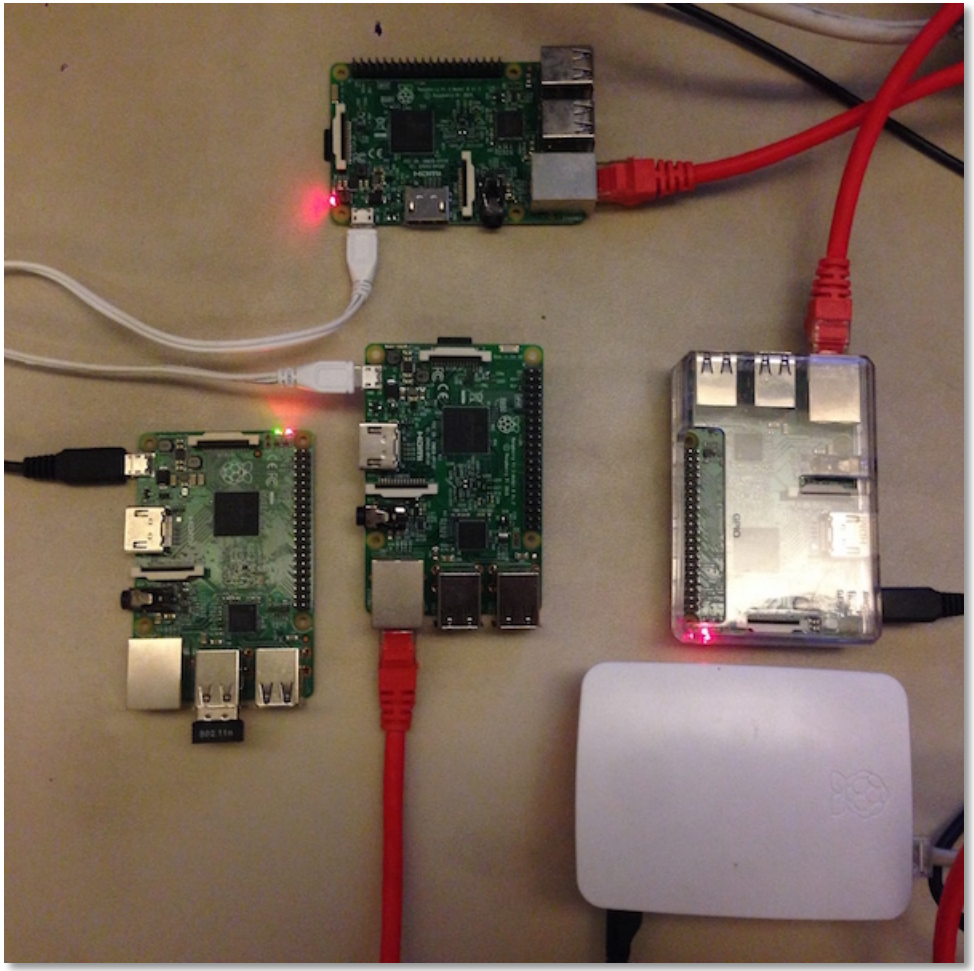


Hacking > Crash Dump

```
Microsoft (R) Windows Debugger Version 10.0.10586.567 X86
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [d:\winiot\WebManagement.exe-LiveUM-2016-07-24-12-36-09.dmp]
User Mini Dump File: Only registers, stack and portions of memory are available

Symbol search path is: srv*
Executable search path is:
Windows 10 Version 14376 MP (4 procs) Free ARM (NT) Thumb-2
Product: WinNt, suite: SingleUserTS
Built by: 10.0.14376.0 (rs1_release.160624-1700)
Machine Name:
Debug session time: Mon Jul 25 03:36:09.000 2016 (UTC + 8:00)
System Uptime: not available
Process Uptime: 1 days 4:48:37.000
.............................................................
........
Loading unloaded module list
.
Cannot read PEB32 from WOW64 TEB32 ffffffff - Win32 error 0n30
Unable to load image C:\Windows\System32\ntdll.dll, Win32 error 0n2
*** WARNING: Unable to verify timestamp for ntdll.dll
ntdll!NtWaitForSingleObject+0x6:
*** WARNING: Unable to verify timestamp for KERNELBASE.dll
77320ab6 4770     bx          lr {KERNELBASE!WaitForSingleObjectEx+0xc0 (76fedf30)}
0:000> |
.  0         id: 698       examine       name: C:\Windows\System32\WebManagement.exe
0:000> !peb
PEB at 032f8000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         00a00000
    Ldr                       773eb9e0
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 034a1730 . 034ae758
    Ldr.InLoadOrderModuleList:           034a1808 . 034ae748
    Ldr.InMemoryOrderModuleList:         034a1810 . 034ae750
            Base TimeStamp                     Module
          a00000 576dee48 Jun 25 10:36:56 2016 C:\windows\system32\WebManagement.exe
        77300000 576deb18 Jun 25 10:23:20 2016 C:\windows\SYSTEM32\ntdll.dll
        76f20000 576debe7 Jun 25 10:26:47 2016 C:\windows\System32\KERNELBASE.dll
        770b0000 576debda Jun 25 10:26:34 2016 C:\windows\System32\combase.dll
        76ce0000 576deb16 Jun 25 10:23:18 2016 C:\windows\System32\ucrtbase.dll
        76e30000 576ded32 Jun 25 10:32:18 2016 C:\windows\System32\RPCRT4.dll
        76de0000 576dee1b Jun 25 10:36:11 2016 C:\windows\System32\kernel32legacy.dll
        76d90000 576deeaa Jun 25 10:38:34 2016 C:\windows\System32\bcryptPrimitives.dll
```

Hacking > Fuzzing

- Current Approach
  - Old school
  - REST APIs to control device
- Future Approach
  - Corpus driven fuzzing
  - WinAFL



Recommendations

Recommendations

Segment your network	• Mitigates lateral infection • Incident isolation and cleanup
Protect network services	• Use built-in firewall • Disable unnecessary services
Change default Administrator password	• Eliminates most malware infection attempts today
Use devices supporting TPM	• Minnowboard +Dragonboard • Raspberry Pi + Discrete TPM
Take advantage of available security features	• Enable Secure Boot • Enable BitLocker

Conclusion

Conclusion

Windows 10 IoT Core's features makes it an attractive alternative to today's IoT OS

Attack surface is smaller than other computing devices, but if IoT services are factored in, will be bigger

Vendors/makers should be careful about mis-configurations

More security research needed/encouraged

Questions?

IBM Security

THANK YOU

FOLLOW US ON:

- ibm.com/security
- securityintelligence.com
- xforce.ibmcloud.com
- @ibmsecurity
- youtube/user/ibmsecuritysolutions

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective.

IBM DOES NOT WARRANT THAT ANYSYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

IBM