Monthly Research

Security of Windows 10 IoT Core

FFRI, Inc.
http://www.ffri.jp

Ver 1.00.01

Introduction

- Windows 10 IoT is successor platform of Windows Embedded that optimized for embedded devices.

- Windows 10 IoT Core Insider Preview has been provided for single-board computers such as the Raspberry Pi 2.

- We show tutorial about security of Windows 10 IoT Core using the Raspberry Pi 2.

  * This report describes about preview version of Windows 10 IoT Core. So it may be different in official release.

Difference between Windows 10 Editions

- Windows 10 has 7 editions.
  - Home, Mobile, Pro, Enterprise, Education, Enterprise Mobile, Windows 10 IoT
- Windows 10 Mobile supports ARM and x86/x64.
- Differences between major editions and Windows 10 IoT are footprint and hardware control API for GPIO(Windows.Devices API).
- Windows 10 IoT are targeting next generation embedded devices.

Editions of Windows 10 IoT

Edition	Description	System Requirements
Windows 10 IoT for Industry Devices	Only works on x86/x64. It is High-function OS which has Desktop Shell.	RAM：1 GB Storage：16 GB
Windows 10 IoT for Mobile Devices	Only works on ARM 32bit. Successor of Windows Embedded Handheld OS. It has Modern Shell.	Mobile device RAM：512 MB Storage：4 GB
Windows 10 IoT for Small Devices, Windows 10 IoT Core	Works on x86 and ARM 32bit. Lightweight OS for single-board computers. No Shell	RAM：256MB Storage：2GB

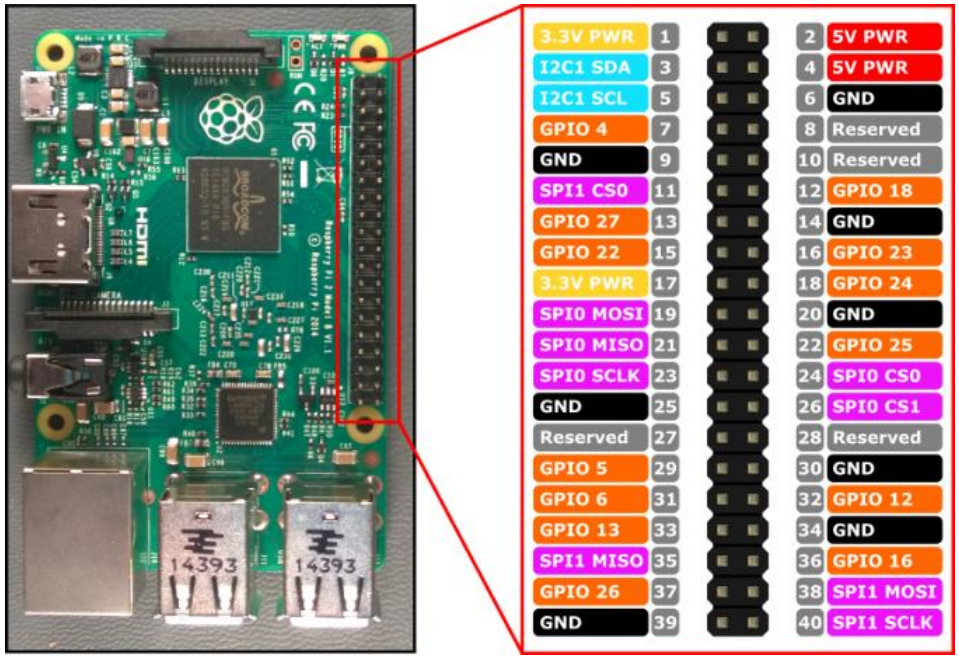
Supported Single-Board Computers

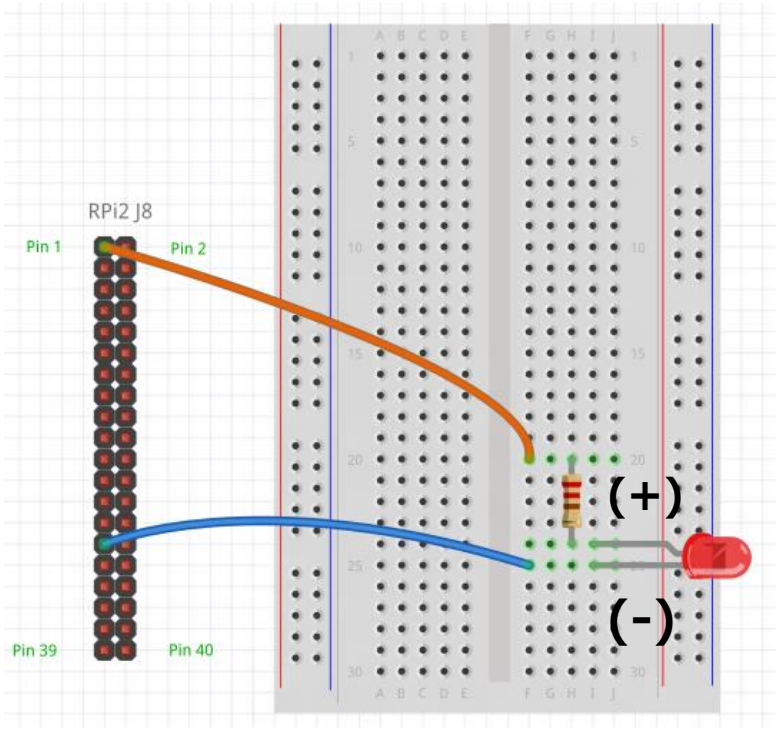
Product	CPU	RAM
Raspberry Pi 2	900MHz quad-core ARM Cortex-A7 CPU	1GB
MinnowBoard Max	Atom E3815-1.46GHz/E3825-1.33GHz	1 or 2GB
Galileo	Intel® Quark™ SoC X1000 (16K Cache, 400 MHz)	256 MB
Windows Remote Arduino	ATmega2560 16 MHz	256 KB
Windows Virtual Shields for Arduino	ATmega328 16 MHz	32 KB

- Windows 10 IoT is likely to be spread by supporting popular single-board computer such as the Raspberry Pi 2.
- It's possible to develop IoT applications using C# or C++ or Python.

Blinky with Windows 10 IoT Core for Raspberry Pi 2

- Blinking a LED by controlling GPIO using Windows 10 IoT Core for Raspberry Pi 2.





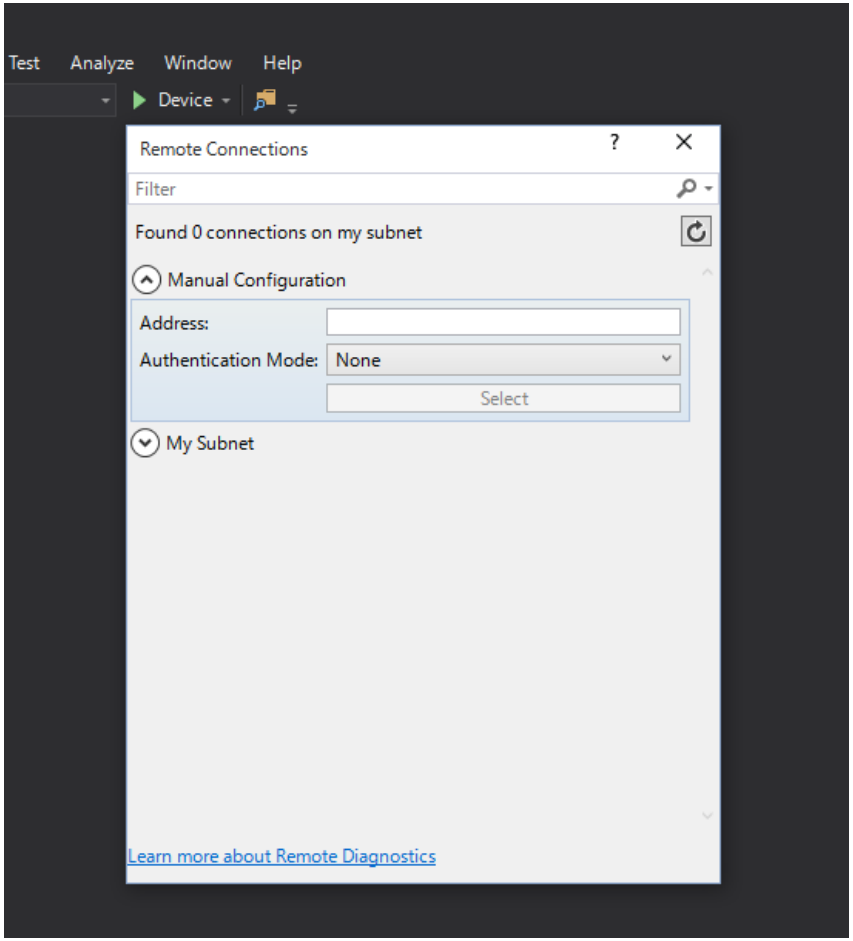
ms-iot.github.io

Blinky Steps on Windows 10 IoT Core

1. Install and setup Visual Studio 2015 RC on your PC.
2. Install the Python 3.x, PTVS(Python Tools for Visual Studio), Python UWP(Universal Windows Platform) SDK.
3. Building the circuit shown in previous slide using a red LED, a 200Ω resistor, a breadboard and jumper wires.
   Make sure the shorter leg (-) is connected to GPIO 5 and the longer leg (+) to the resistor or it won't light up.
4. Power on the Raspberry Pi 2 and connect the same network as PC.
5. Checking Raspberry Pi 2 IP through Windows 10 IoT Core Watcher included Visual Studio.
6. Writing the Blinky program and deploy it from "**Remote Machine**" on "Device button" menu.

Deploying from Visual Studio

- Setting the remote deploy from visual studio "Device button“ menu.
- We can deploy the program through select “None” on “Authentication Mode” setting on Visual Studio.



Blinky Python Script

```
import _wingpio as gpio       // Import the module control the GPIO
import time

led_pin = 5                   // Set PIN number of GPIO
ledstatus = 0

gpio.setup(led_pin, gpio.OUT, gpio.PUD_OFF, gpio.HIGH)

while True:
    if ledstatus == 0:
        ledstatus = 1
        gpio.output(led_pin, gpio.HIGH)     // Set the specified GPIO of PIN to
HIGH(On)
    else:
        ledstatus = 0
        gpio.output(led_pin, gpio.LOW)      // Set the specified GPIO of PIN to
LOW (Off)

    time.sleep(0.5)    // It specifies the interval of Blink
gpio.cleanup()
```

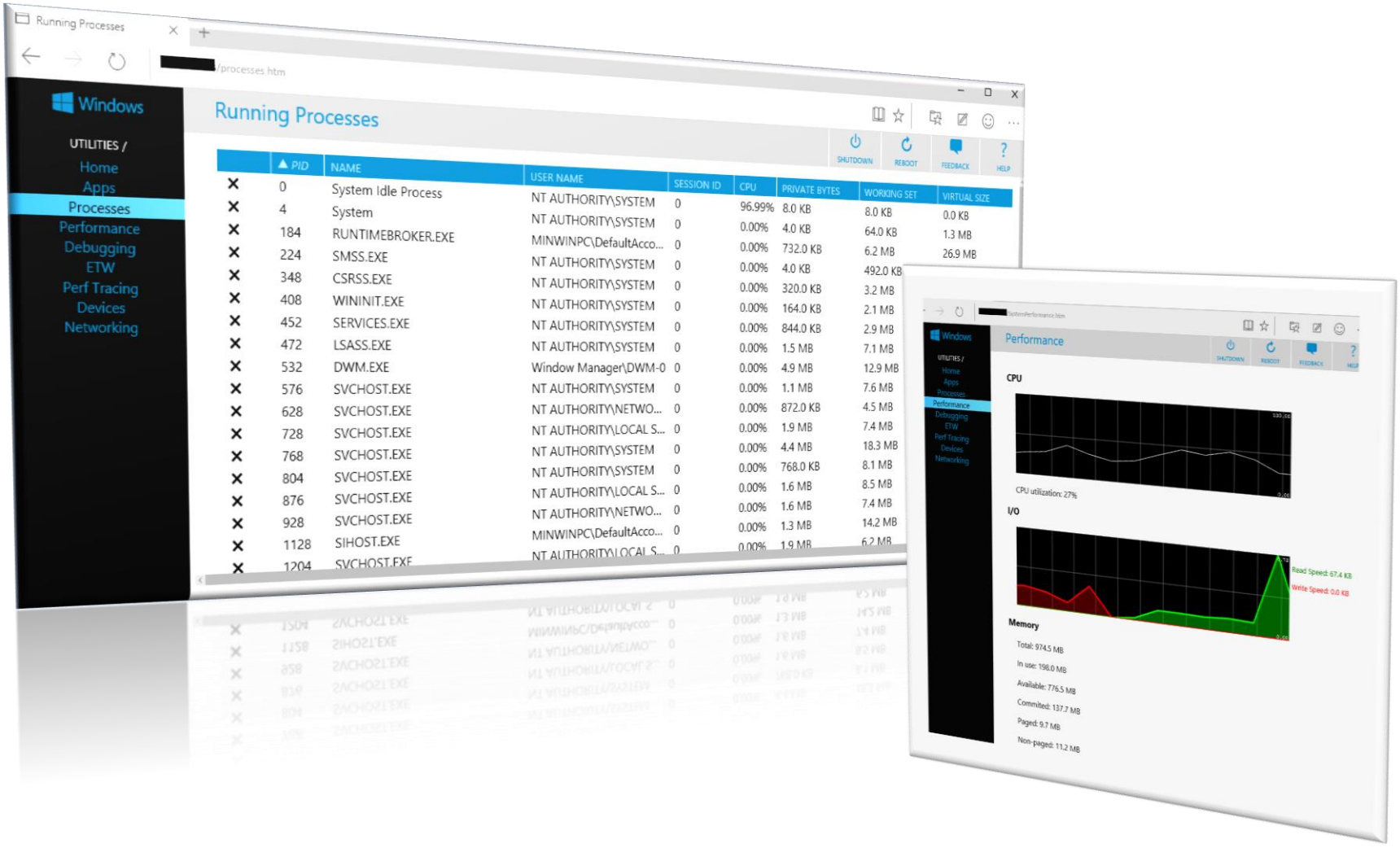
Summary of Blinky

- Windows 10 IoT Core can control hardware via Windows.Devices API.

- You can develop embedded apps easily with popular programing languages like Python and C# on Visual Studio. You can also develop GUI app.
  - http://ms-iot.github.io/content/en-US/win10/samples/HelloWorld.htm

- But we can deploy the program through select “None” on “Authentication Mode” setting on Visual Studio.

Web Interface



Web Interface

- Some functions are available via web interface.
  "http://<Device's IP>" (Need Basic Authentication).

Menu	Description
Apps	Install, Uninstall, Start application. Checking running applications.
Processes	Getting list of running processes.
Performance	Checking status of CPU, RAM, I/O on real-time.
Debugging	Getting kernel dump or process dump. Setting and viewing crash dumps.
ETW	Checking events.
Perf Tracing	Checking memory leak using WPR(Windows Performance Recorder).
Devices	Checking connected devices on single-board.
Networking	Checking the network status.

Port Scanning

- Some services has detected by Nmap.
- These services are working by Windows 10 IoT Core default settings.

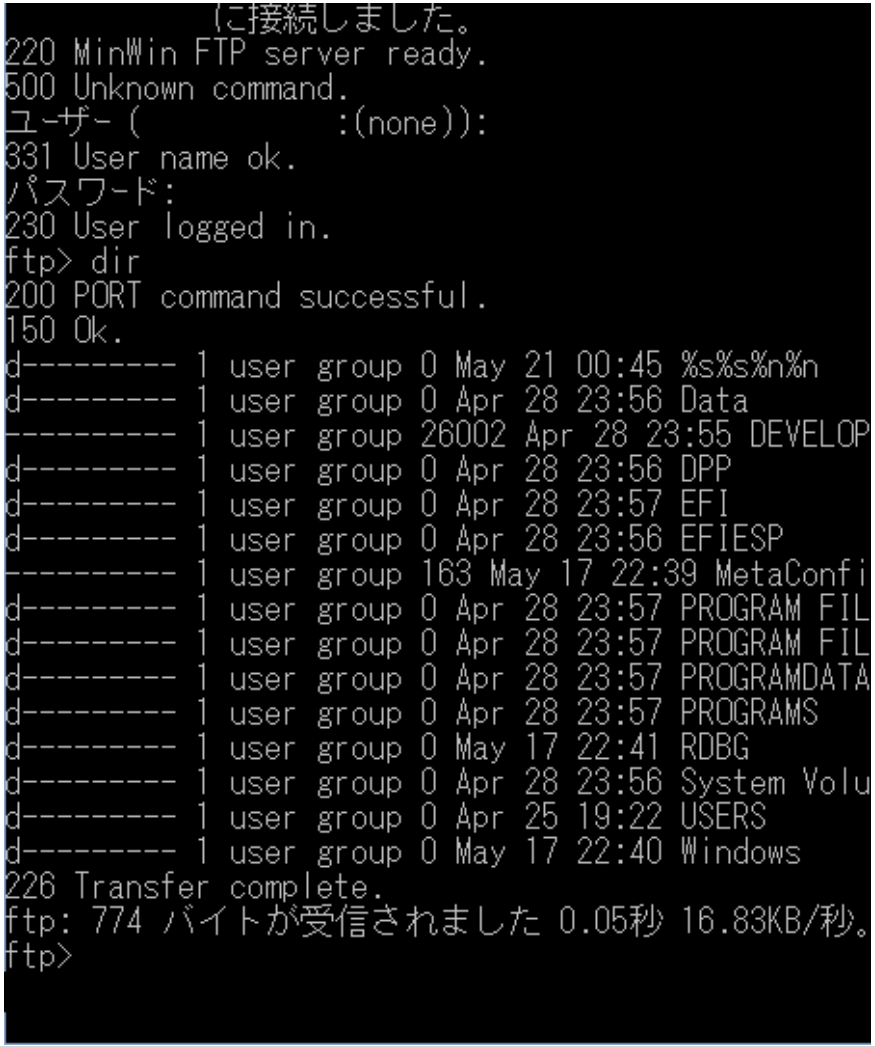
PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	
22/tcp	open	ssh	(protocol 2.0)
80/tcp	open	http	Microsoft-HTTPAPI/2.0
135/tcp	open	msrpc	Microsoft Windows RPC
445/tcp	open	microsoft-ds?	
4020/tcp	open	trap	
5985/tcp	open	wsman	
9956/tcp	open	unknown	
29817/tcp	open	unknown	
29819/tcp	open	unknown	
29820/tcp	open	unknown	

Port Scanning

- Nmap detected Windows series by OS finger printing.
- SSH and HTTP require authentication. But FTP does not require authentication.
- This ssh service provides command execution only.
- File Transfer(SCP, SFTP), port forwarding and Public key authentication are maybe not supported.

World Readable FTP

- Any username and password (incl. null) will be accepted.
- FTP Root directory is “/” and it is also root of file system. You can read all directories and files. But you can not write.
  - You can create and delete any files if it is mounted as network share drive with valid authentication.(FAT)
- It have a low possibility of malware infection via FTP. But an attacker can read any file on the device.



Security functions of Windows 10 IoT Core

- Windows Firewall is disable by default.
  You cannot configure it via web interface. But you can use "netsh" command.
- Windows Update is not provided.
- Windows Defender is not provided.
- UAC is disabled.
  - Maybe it's not necessary because Windows 10 IoT Core doesn't have interactive UI.
- DEP and ASLR are enabled by default.
- Control Flow Guard is supported. But you need to configure linker option for your project.
  - Project>Property>C/C++>Code Generation>Control Flow Guard

Start-up programs (startup.exe)

```
C:¥>startup
startup
Startup Editor

Options:
  /d - display the list of startup apps
  /r <Name> - remove an app from the list of startup apps
  /a <Name> <Command> - add an app into the list of startup apps

Where:
  <Name> is the name of the app in the startup registry
  <Command> is the full command line for the app

Example:
  Startup /a EbootPinger "start ¥windows¥system32¥EbootPinger.exe"

C:¥>
```

- Start-up program is possible to display (/d), remove(/r), add (/a) through "C:¥Windows¥system32¥STARTUP.EXE".
- ftpd.exe had been added by default.

Threat Analysis

- Illegal access
  - Many users will not change the default password of built-in administrator account because there is no setup wizard about password.
  - It is easy to hijack the devices through web interface using default password.
  - Attacker will develop auto attacking tool for device that use default password.
- Password Cracking
  - Attacker might attempt to crack ftp, ssh, http authentication.
- Account steal by sniffing packets
  - Attacker will attempt access to admin interface if password is leaked. Because FTP and HTTP are vulnerable to sniffing.

Threat Analysis (cont'd)

- Leak of data
  - “Windows IoT Core Watcher” installed with Visual Studio can find Windows IoT devices on the same network.
  - So, Attacker will get some data or programs through FTP service that running by default and does not require authentication.

- Hardware hijack
  - Attacker can control hardware(camera, switch, etc.) on Single-board because Visual Studio does not require authentication and controlling GPIO is easy through Windows.Devices API.

Threat Analysis (cont'd)

- Tampering of data, malware infection
  - Worm that exploits Visual Studio's deploy function might be outbreak.
  - Attacker would add malware to startup through registry or "STARTUP.EXE" if he is able to execute arbitrary OS commands.
  - Attack scenario
    1. RAT infects a PC which is connected to the same network as target device.
    2. Searching Windows 10 IoT device through the PC.
    3. Execute backdoor program through Visual Studio.
    4. Attacker gets persistent control by uploading malware or overwriting original programs.
    5. Attacker gets the control about hardware through Windows.Devices API.

Countermeasures

- There are countermeasures for the threats described above.
  - Add admin user and set strong password.
    - Setting by command like "net user <username> /add" through remote shell.
  - Set the rules on the network using firewall.
    - Limit the connection port and IP through "netsh" command.
  - Stop the unnecessary service.
    - Stop FTP if you don't need it.
  - Secure Communication.
    - Use SSH for maintenance. Do not use web interface via internet.
    - Use strong encryption than WEP for Wi-Fi.
  - Physical security
    - Protect I/O interface and validate signals from GPIO in application.

References

- Introducing Windows 10 Editions
  http://blogs.windows.com/bloggingwindows/2015/05/13/introducing-windows-10-editions/
- WinHEC Shenzhen 2015
  https://channel9.msdn.com/Events/WinHEC/2015
- Internet of Things Overview(Build 2015)
  https://channel9.msdn.com/Events/Build/2015/2-652
- Python Tools for Visual Studio
  https://pytools.codeplex.com/
- Windows IoT - Python Blinky Sample
  https://ms-iot.github.io/content/en-US/win10/samples/PythonBlinky.htm
- Windows IoT - Blinky Sample
  https://ms-iot.github.io/content/en-US/win10/samples/Blinky.htm
- Nmap
  https://nmap.org/

Contact Information

E-Mail ： research—feedback@ffri.jp
Twitter： @FFRI_Research