Ghost Tunnel V2
Covert Data Exfiltration Channel to Circumvent Air Gapping

Yongtao Wang, Kunzhe Chai, Mingchuang Qin
PegasusTeam, 360 Security Technology
May 5, 2019
360 Security Technology is a leading Internet security company in Asia. Our core products are anti-virus security software for PC and cellphones.

**Pegasus** is a red team from 360 Security Technology focusing on wireless and IoT Security, we created 360SkyScan WIPS, we have achieved 100% success rate in our wireless pentest, our team was founded in 2015.
Agenda

- Introduction
- Previous research on Air-Gapped attack
- Ghost Tunnel V1 revision
- Ghost Tunnel V2 Introduction
- Ghost Tunnel V2 implementation
Introduction

• Air-Gapping
• Attack events
Air Gapping

• Air gapping
  - Wikipedia: “air gapping[1] is a network security measure employed on one or more computers to ensure that a secure computer network is physically isolated from unsecured networks, such as the public Internet or an unsecured local area network.[2] The name arises from the technique of creating a network that is physically separated (with a conceptual air gap) from all other networks.”

• Air gapping aims to avoid the intrusion and data leakage through network connections
Air-Gapped Network

• Considered to be the most secure
Nothing Is Impossible

• Attack Vectors
  - Malicious USB
  - Employee's laptop
Stuxnet Worm (2010)

- Attacking initiated via an infected USB drive
- Designed to sabotage centrifuges used at a uranium enrichment plant in Iran
NSA Leaks (2013)

- **COTTONMOUTH-I**
  - A USB hardware implant
  - Air-Gap bridging
  - Extracting data from targeted systems via RF signals
Previous research on Air-Gapped attacks
Previous research - 1

- Using radio frequencies to transmit data from a computer
  - Computer monitor
  - Mobile phone FM radio receiver

url: https://thehackernews.com/2014/10/airhopper-hacking-into-isolated.html
Previous research - 2

- A covert bi-directional communication channel between two close by air-gapped computers communicating via heat

url: https://thehackernews.com/2015/03/hacking-air-gapped-computer.html
Previous research - 3

• Data exfiltration via RF signal by attacking Siemens PLCs

Ghost Tunnel V1 Revision

A Covert Data Exfiltration Channel Using Wi-Fi
Ghost Tunnel V1 Revision

• A covert WiFi channel using Beacon, Probe Request, Probe Response
• A special SSID as the identifier
Ghost Tunnel V2

A Covert Data Exfiltration Channel Using Bluetooth Low Energy
Air-gapped Attack

- Implant
  - Malicious software/hardware

- A covert communication channel
  - Any medium that can carry data is possible
Ghost Tunnel V2

**Implant malware**
- USB HID attack
- BashBunny

**Setup C&C tunnel**
- Via BLE Adv

**Exfiltrate data**
- Execute Command
Ghost Tunnel V2

- Can bypass firewalls
- Cross-Platform support
- Effective range up to 100 meters (@20dBm)
The Usual Bluetooth Connection Process

Initiator

Responder

Established LL connection

(Optional) Security_Request

Pairing_Request

Pairing_Response

Phase 1

Pairing over SMP:
Legacy pairing or Secure Connections

Phase 2

Establishment of encrypted connection with key generated in phase 2

Key Distribution

Key Distribution

Key Distribution

Phase 3
Ghost Tunnel V2 – No Bluetooth Connection

Host A ➔ Scanner ➔ Advertiser ➔ Host B

- **Discovery**:
  - ADV_IND: Ch37
  - ADV_IND: Ch38
  - ADV_IND: Ch39

- **Connecting**:
  - ADV_IND: Ch38
  - CONNECT_REQ: Ch38

- **Master** ➔ **Slave**
  - DATA_TX: Ch11
  - DATA_RX: Ch11

- **GAP “Central” Role**
- **GATT “Client” Role**

- **GAP “Peripheral” Role**
- **GATT “Server” Role**

Unassigned/Standby

**Time** ➔ **Unassigned/Standby**
Bluetooth Low Energy State

Host A

Scanner

Adv_IND: Ch37
Adv_IND: Ch38
Adv_IND: Ch39

Initiator
Adv_IND: Ch38
Connect_REQ: Ch38

Master
Data_TX: Ch11
Data_RX: Ch11

Slave

Host B

Advertiser

Unassigned/Standby

Discovery

Connecting

GAP “Central” Role
GATT “Client” Role

GAP “Peripheral” Role
GATT “Server” Role

Connected
# Bluetooth Low Energy Frames

<table>
<thead>
<tr>
<th>Discovery</th>
<th>Connection</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADV_IND</td>
<td>CONNECT_IND</td>
<td>DATA_TX</td>
</tr>
<tr>
<td>SCAN_REQ</td>
<td>CONNECT_REQ</td>
<td>DATA_RX</td>
</tr>
<tr>
<td>SCAN_RSP</td>
<td>CONNECT_RSP</td>
<td>......</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>
Passive Scanning for BLE Networks

Scanner

ADV_IND Packet: Ch37
ADV_IND Packet: Ch38
ADV_IND Packet: Ch39
...
...
...
ADV_IND Packet: Ch37
ADV_IND Packet: Ch38
ADV_IND Packet: Ch39
...
...
...

Advertiser

Advertising Event

Advertising Interval

Advertising Event

time
Active Scanning for BLE Networks

Scanner

ADV_IND Packet: Ch37
ADV_IND Packet: Ch38
ADV_IND Packet: Ch39

Advertiser

Advertising Data

Advertising Interval

SCAN_REQ Packet: Ch37
SCAN_RSP Packet: Ch37

Scan Response Data
Ghost Tunnel V2 Implementation
Bluetooth Low Energy Packet

BLE Packet

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Access Address</th>
<th>Protocol Data Unit (PDU)</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>4 Bytes</td>
<td>2-257 Bytes</td>
<td>3 Bytes</td>
</tr>
</tbody>
</table>

Advertising Channel PDU

- Header: 2 Bytes
- Payload: 0-37 Bytes

Data Channel PDU

- Header: 2 Bytes
- Payload: up to 255 Bytes (incl. MIC)
- MIC*: 4 Bytes

Ref: BT Specification v4.2, Vol. 6, Part B, Sec. 2.1

*Message Integrity Check: Included as part of Payload if used (for security)
Advertising Channel PDU

---

**Advertising Channel PDU**

<table>
<thead>
<tr>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Bytes</td>
<td>6-37 Bytes</td>
</tr>
</tbody>
</table>

---

**Advertising Packet Payload**

- **ADV Address**
- **AD0 Structure**
- **...**
- **ADN Structure**

---

**Unknown**

- Length: 12
- Type: Unknown (0xa)
- Data: 637573746f6d2064617461
### Advertisement Data Structures

#### Advertisement Data

<table>
<thead>
<tr>
<th>ADV Data</th>
<th>AD Length</th>
<th>AD Type</th>
<th>AD Data</th>
</tr>
</thead>
</table>

This table represents the structure of advertisement data with columns for ADV Data, AD Length, AD Type, and AD Data.
Key Problem

• How to send and receive Bluetooth Low Energy data frames through local Bluetooth interface in user space?

• Bluetooth interface mode
  - BR/EDR (audio…)
  - BLE (IoT, wearable devices…)
  - …
Data Format

<table>
<thead>
<tr>
<th>Identify</th>
<th>length</th>
<th>type</th>
<th>Company id</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x05</td>
<td>0xFF</td>
<td>0xFFF</td>
<td>0xFFFFE</td>
<td>0x1234</td>
</tr>
<tr>
<td>Custom payload</td>
<td>len</td>
<td>type</td>
<td>Custom data</td>
<td></td>
</tr>
</tbody>
</table>
Send BLE Data

```csharp
class BLEData
{
    public void SendData(string buf)
    {
        var publisher = new BluetoothLEAdvertisementPublisher();
        var manufacturerData = new BluetoothLEManufacturerData();
        manufacturerData.CompanyId = 0xFFFE;
        var writer = new DataWriter();
        writer.WriteUInt16(0x1234);
        manufacturerData.Data = writer.DetachBuffer();
        publisher.Advertisement.ManufacturerData.Add(manufacturerData);
        var data = new BluetoothLEAdvertisementDataSection();
        writer.WriteString(buf);
        data.Data = writer.DetachBuffer();
        data.DataType = 0x1aa;
        publisher.Advertisement.DataSections.Add(data);
        publisher.Start();
    }
}
```

<table>
<thead>
<tr>
<th>length</th>
<th>type</th>
<th>Company id</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x05</td>
<td>0xFF</td>
<td>0xFFFE</td>
<td>0x1234</td>
</tr>
<tr>
<td>sizeof(type+buf)</td>
<td>0x1aa</td>
<td>buf</td>
<td></td>
</tr>
</tbody>
</table>
Receive BLE Data

```java
public void RecvData()
{
    var watcher = new BluetoothLEAdvertisementWatcher();
    var manufacturerData = new BluetoothLEManufacturerData();
    manufacturerData.CompanyId = 0xFFFE;
    var writer = new DataWriter();
    writer.WriteByte(0x1234);
    manufacturerData.Data = writer.DetachBuffer();
    watcher.AdvertisementFilter.Advertisement.ManufacturerData.Add(manufacturerData);
    watcher.SignalStrengthFilter.InRangeThresholdInDBm = -90;
    watcher.SignalStrengthFilter.OutOfRangeThresholdInDBm = -95;
    watcher.SignalStrengthFilter.OutOfRangeTimeout = TimeSpan.FromMilliseconds(2000);
    watcher.Received += OnAdvertisementReceived;
    watcher.Start();
}
```

<table>
<thead>
<tr>
<th>length</th>
<th>type</th>
<th>Company id</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x05</td>
<td>0xFF</td>
<td>0xFFFE</td>
<td>0x1234</td>
</tr>
<tr>
<td>sizeof(type+buf)</td>
<td>0xaa</td>
<td>buf</td>
<td></td>
</tr>
</tbody>
</table>
Ghost Tunnel V2– No Connection

- A covert BLE channel using ADV_IND, SCAN_REQ, SCAN_RSP.
- A special Custom manufacture ID as the identifier
Thanks! & QA?
github@360pegasusteam