DISCOVERING AND EXPLOITING NOVEL SECURITY VULNERABILITIES IN APPLE ZEROCONF

(Xiaolong Bai, Luyi Xing) (co-first authors),
Nan Zhang, XiaoFeng Wang, Xiaojing Liao, Tongxin Li, Shi-Min Hu

Tsinghua University,
Indiana University Bloomington
Georgia Institute of Technology,
Peking University
Who are we?

• System Security Lab, Indiana University Bloomington
  – Focus on novel problems in system security

• Our advisor: Prof. XiaoFeng Wang
  – Top 10 authors on leading security venues for the past 10 years
  – http://www.informatics.indiana.edu/xw7/
Who are we?

• We have two talks on Black Hat USA 2016
  – Luyi Xing and Xiaolong Bai, DISCOVERING AND EXPLOITING NOVEL SECURITY VULNERABILITIES IN APPLE ZEROCONF, August 4, Jasmine Ballroom, 12:10 - 13:00
  – Nan Zhang, DANGEROUS HARE: HANGING ATTRIBUTE REFERENCES HAZARDS DUE TO VENDOR CUSTOMIZATION, August 4, South Seas GH, 17:00 - 17:25
DISCOVERING AND EXPLOITING NOVEL SECURITY VULNERABILITIES IN APPLE ZEROCONF
ZeroConf

• Zero Configuration Networking
• Automatically configures a usable computer network
  – No manual configuration
  – No specific configuration server
• Designed to reduce users’ burden
  – Setting up a new network
  – Use a new service.
ZeroConf

• Bonjour protocol
  – zero-configuration networking over IP that Apple has submitted to the IETF.

• Goals:
  – With little or no configuration
  – to add devices/services to a local network
  – Existing devices can automatically find and connect to those new devices/services
Bonjour

• Administrators
  – no need to assign IP, host names, service names to network services (e.g., printer)

• When using a service, users simply
  – ask to see what network services are available
  – and choose from the list of automatically discovered services.
How about traditional configured network?
Traditionally

Must Configure:

- IP
- Printer name,
  - e.g., lh135-soic.ads.iu.edu
- DNS server
Traditionally

Must Configure:

- IP
- Printer name,
  - e.g., lh135-soic.ads.iu.edu
- DNS server
Features of Bonjour

1. Service configures itself
   – IP, hostname, service instance name

2. Clients automatically discover available services
   – No pre-knowledge of the service’s name, hostname or IP
1. ZeroConf Concept

2. So, how?
Features of Bonjour

1. Service configures itself
   – IP, hostname, service instance name

2. Clients automatically discover available services
   – No pre-knowledge of the service’s name, hostname or IP
Add a new printer to a network
Is anybody using IP fe80::abcd:1234....?
A printer configures itself

No?
Great, I’ll take it.

IP
fe80::abcd:1234
A printer configures itself

Anybody using hostname HP9FE5.host.local?
A printer configures itself

No?
Wonderful, I’ll take it.

IP
fe80::abcd:1234
Hostname
HP9FE5.host.local
A printer configures itself

Anybody having a printing service named HP-Service-9FE5?

IP
fe80::abcd:1234
Hostname
HP9FE5.host.local
Service Instance Name
HP-Service-9FE5
A printer finishes configuring itself
Features of Bonjour

1. Service configures itself
   - IP, hostname, service instance name

2. Clients automatically discover available services
   - No pre-knowledge of the service’s name, hostname or IP

Two phases: Discovery and Resolution
Q1:
Anyone has a printer service?

A1:
I have HP-Service-9FE5
Automatically find the printer: Resolution

Q1:
Anyone has a printer service?

Q2:
So on which host is this HP-Service-9FE5?

A1:
I have HP-Service-9FE5

A2:
It’s on host HP9fe5.host.local
Automatically find the printer: Resolution

Q1: Anyone has a printer service?

Q2: So on which host is this HP-Service-9FE5?

Q3: What is the address of NPI9fe5.host.local?

A1: I have HP-Service-9FE5

A2: It’s on host HP9fe5.host.local

A3: Its address is fe80::abcd:1234
Added/Saved the printer to your list

IP: fe80::abcd:1234
Hostname: HP9FE5.host.local
Service Instance Name: HP-Service-9FE5
Added/Saved the printer to your list

Apple:

Applications store service instance names, so if the IP, port, or host name changed, the application can still connect.
Service instance name `HP-Service-9FE5` is saved

Saved printer = A printer who owns service name `HP-Service-9FE5`
Adversary Model

• On a device (malware infected) in your local network
• Aims to intercept secrets/files transferred between uninfected devices
Adversary Model

• Your Mac/printer are un-infected
• Steal your printing documents?
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking

Case 1: Attack Bonjour
Attack Bonjour

• Two examples
• Printer
  – Printers using Bonjour
• PhotoSync
  – Synchronizing photos between Mac and iPhone using Bonjour
• Not an application-specific or service-specific problem
  – Vulnerabilities in the design of Bonjour protocol
A device infected by malware

IP
Hostname
Service Instance Name
HP-Service-9FE5
A device infected by malware

I have a printing service instance named HP-Service-9FE5
I have a printing service instance named **HP-Service-9FE5**
Saved printer =
A printer who owns service name **HP-Service-9FE5**
Why it happens?

Three **Changing** Attributes:

- IP
- Hostname
- Service Instance Name

Apple:

*Applications store service instance names, so if the IP, port, or host name changed, the application can still connect.*
Lack of authentication

Three Changing Attributes:

- IP
- Hostname
- Service Instance Name

- Anyone can claim any value of the three attributes
- The protocol only guarantees no duplicates.
If not saving service instance names, is it secure enough?

No!
Attack Bonjour

• PhotoSync
  – Synchronizing photos between Mac and iPhone using Bonjour

• Not saving service instance name
  – Client discovers and resolves the server each time
Normally

• Discovery: Client browses for server

Who has PhotoSync service

Client

Server

--- means broadcast
Normally

- Discovery: Server responds with service instance name

Who has PhotoSync service

I have. service instance name: abcd

Client

Server

---

Means broadcast
Normally

- Resolution 1: Client queries for the host name of the service

Who has PhotoSync service

I have. service instance name: abcd

What is the host name of abcd

---

Client

Server

---

means broadcast
Normally

• Resolution 1: Server responds with the host name

Who has PhotoSync service
I have. service instance name: abcd
What is the host name of abcd
Its host name is Macbook

Client

Server

--- means broadcast
Normally

• Resolution 2: Client queries for the address of the host

Who has PhotoSync service
I have. service instance name: abcd
What is the host name of abcd
Its host name is Macbook
What is the address of Macbook

Means broadcast
Normally

• Resolution 2: Server responds with its address

Who has PhotoSync service
I have. service instance name: abcd
What is the host name of abcd
Its host name is Macbook
What is the address of Macbook
Its address is 192.168.0.1

Client

Server

means broadcast
What Can Go Wrong?

• Another malware-infected device spoofs the client
  – Successful Man-in-the-Middle
• During Resolution
  – Service instance name to host name
  – Host name to address
What Can Go Wrong?

- Attack 1: service instance name to host name
What Can Go Wrong?

- Attack 1: service instance name to host name

The host name of service instance abcd is Macbook

The host name of service instance abcd is Mallory
What Can Go Wrong?

• Attack 1: service instance name to host name
What Can Go Wrong?

• Attack 1: service instance name to host name
What Can Go Wrong?

• Attack 2: service instance name to host name

What is the address of host Macbook
What Can Go Wrong?

• Attack 2: service instance name to host name

The address of host Macbook is 192.168.0.1

The address of host Macbook is 192.168.0.100
What Can Go Wrong?

• Attack 2: service instance name to host name
What Can Go Wrong?

- Attack 2: service instance name to host name
Demo

- [https://www.youtube.com/watch?v=WUWusqgqFr0&feature=youtu.be](https://www.youtube.com/watch?v=WUWusqgqFr0&feature=youtu.be)
Fundamental Problem

• Lack of authentication
• Anyone can claim any value of the identification attributes
• The protocol only guarantees no duplicates, but not security.

Is it easy to provide authentication?

No!
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking

Case 2: Airdrop
Airdrop between Apple devices

• With AirDrop, you can share photos, videos, websites, locations, and more with people nearby with an Apple device.
Jeff’s Macbook:
Q1: Anyone has an airdrop service?

Alice’s iPhone:
I have a service named abcd.airdrop.service
Attack Airdrop

Jeff’s Macbook:
Q2: So **on which host** is Alice’s service?
Attack Airdrop

Jeff’s Macbook:
Q2: So on which host is Alice’s service?

Alice’s iPhone:
A2: It’s on host Alices.iphone.local

Bob’s iMac:
A2: It’s on host Bobs.imac.local
Alice’s iPhone has service named abcd.airdrop.tcp, which is on host Bobs.imac.local

Jeff’s Macbook:
Q2: So on which host is Alice’s service?

Alice’s iPhone:
A1: It’s on host Alices.iphone.local

Bob’s iMac:
A2: It’s on host Bobs.imac.local
Does TLS help?

Jeff’s Macbook:
Connect
https://Bobs.imac.local

Alice’s iPhone:
A2: It’s on host Alices.iphone.local

Bob’s iMac:
A2: It’s on host Bobs.imac.local
TLS in Airdrop

https://Bobs.imac.local

Server certificate issued to appleid. CDEF ...

https://Alices.iphone.local

Server certificate issued to appleid.ABCD...

Bob’s iMac

Alice’s iPhone
So the certificate in airdrop can hardly be used for authentication.

https://Bobs.imac.local
Server certificate issued to appleid.CDEF...

https://Alices.iphone.local
Server certificate issued to appleid.ABCD...

Jeff’s Macbook

Bob’s iMac

Alice’s iPhone
Domain should match the certificate

https://Bobs.imac.local

Server certificate issued to appleid.CDEF...

https://google.com

Certificate issued to google.com

Jeff’s Macbook

Bob’s iMac
Domain should match the certificate

https://Bobs.imac.local
Server certificate issued to appleid.CDEF...

https://Alices.iphone.local
Server certificate issued to appleid.ABCD...

Bob’s iMac

Alice’s iPhone
What’s wrong with TLS in Airdrop

• The certificate in airdrop cannot be used for authentication
  – E.g., certificate should be issued to Alice
  – but indeed issued to appleid.ABCD...

• The certificate should be issued to WHAT?
What’s wrong with TLS in Airdrop

• Issue the certificate to the domain (host name)?
  – No. Host name may change and not representing a user

• Issue the certificate to the user’s name?
  – No. Name can be duplicated

• Issue the certificate to the user’s social security number?
  – No. social security number is too private
What’s wrong with TLS in Airdrop

• Linking a human to her certificate is complicated
  – challenge in finding any identifiable information that are
    • well-known
    • no privacy implication
    • and unique
Demo

- https://www.youtube.com/watch?v=2JEJLpvnRO4
Technical Details

• Airdrop service daemon: /usr/libexec/sharingd
  – Responsible for Bonjour process and https connection

• Not ethernet interface, Apple private interface
  – awdl0: Apple Wireless Direct Link
  – Device-to-device direct link
Technical Details

• How to work on this interface?
  – sharingd uses an Apple-private socket option SO_RECV_ANYIF (0x1104)

```c
on = 1;
status = setsockopt(handle->io_watcher.fd, SOL_SOCKET, 0x1104, &on, sizeof(on));
if(status == -1){
    printf("setsockopt SO_RECV_ANYIF error\n");
}
```
Some customized ZeroConf protocols

• FileDrop
  – TCP packets for discovery
  – elliptic curve cryptography for security
  – Failed in authentication
    • challenge in linking a human to her public key
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking

Case 3: Apple’s Vulnerable framework
Apple’s Vulnerable framework

• Multipeer Connectivity (MC)
  – A framework for automatic service discovery between nearby devices across Wi-Fi and Bluetooth without configuration

• Object to identify each app: peerID
  – displayName (public) & uniqueID (private)
Normally

• Automatic Service Discovery Without Configuration
  – Servers advertise peerIDs
Normally

- Automatic Service Discovery Without Configuration
  - Servers advertise peerIDs, Client browse peerIDs (show displayName)

```
peerID
displayName: Alice
uniqueID: 8573a
```

```
peerID
displayName: Bob
uniqueID: 6c5b3
```
Normally

• Even if servers have the same displayName
Normally

• Even if servers have the same displayName  
  – uniqueIDs generated by MC will always be different

Server

Client
Normally

- Even if servers have the same displayName – uniqueIDs generated by MC will always be different
What Can Go Wrong?

- Attacker acts as both client and server
  - Browse and acquire peerID object from victim server

peerID
displayName: Alice
uniqueID: abcde

Server

Client & Server

Client
What Can Go Wrong?

- Attacker acts as both client and server
  - Advertise using the same peerID object
What Can Go Wrong?

- Client cannot distinguish because of the same uniqueID
What Can Go Wrong?

- Client can not distinguish because of same uniqueID
- Client maps the only peer to attacker’s address (MitM)
Technical Details

• MitM attacker
  – First acts as client browsing for advertising servers
  – Once found a server, advertise using the same peerID

```c
(void)browser:(MCNearbyServiceBrowser *)browser foundPeer:(MCPeerID *)peerID withDiscoveryInfo:
  (NSDictionary *)info {

  ...

  _advertiser = [[MCNearbyServiceAdvertiser alloc] initWithPeer:peerID discoveryInfo:info serviceType:
    _serviceType];
  _advertiser.delegate = self;
  [._advertiser startAdvertisingPeer];

  ...

}
If not using peerID to for identification, is it secure enough?

No!
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking

Case 4: MC in QQ
MC in QQ

- Popular instant messaging software in CN – 829 million active accounts (Wikipedia)
- Face-To-Face Transfer
  - Transfer files between nearby peers by using Multipeer Connectivity
- Not using peerID for identification
  - Customized unique QQ ID
Normally

• Receiver advertises its QQ ID
Normally

• Sender browses for receivers and found their QQ IDs

Sender

Found Receivers:
QQ ID: 1234
QQ ID: 4321

My QQ ID is 1234

Receiver

My QQ ID is 4321

Receiver
Normally

• Sender connects to receiver and gives its QQ ID
Normally

• Sender connects to receiver and gives its QQ ID
What Can Go Wrong?

- Receiver advertises its QQ ID
What Can Go Wrong?

- Attacker found victim receiver’s QQ ID
What Can Go Wrong?

- Attacker advertise using the same QQ ID
What Can Go Wrong?

• Sender found only one QQ ID
What Can Go Wrong?

- Sender connects to Attacker
What Can Go Wrong?

- Attacker connects to Receiver using the Sender’s QQ ID
Demo

• https://www.youtube.com/watch?v=B71FlD3_vrc
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking

Case 5: Bluetooth
All your iOS notifications belong to me

- ZeroConf on Bluetooth: Apple Handoff
  - A service that lets iOS and OS X synchronize data through Bluetooth without configuration
Normally

• Handoff creates Bluetooth Channel without configuration
  – Devices logged in with the same iCloud account
  – Pairing automatically through iCloud account
What Can Go Wrong?

• Bluetooth ZeroConf: No app-level authentication
• Apple Notification Center Service (ANCS)
  – designed for Bluetooth accessories to access notifications on iOS devices
What Can Go Wrong?

- Bluetooth ZeroConf: No app-level authentication
- Apple Notification Center Service (ANCS)
- Through Bluetooth channel created by Handoff
What Can Go Wrong?

• Bluetooth ZeroConf: No app-level authentication
• Apple Notification Center Service (ANCS)
• Through Bluetooth channel created by Handoff
Demo

- [https://www.youtube.com/watch?v=c5viAzAs0Uo](https://www.youtube.com/watch?v=c5viAzAs0Uo)
Summary of attacks

- Attacks on Apple ZeroConf channels
  - Bonjour (Printer, PhotoSync)
  - Airdrop
  - Customized ZeroConf protocols (Filedrop)
  - Multipeer Connectivity (MCBrowserViewController, QQ)
  - Handoff

- All vulnerabilities were reported to vendors, acknowledged by most vendors
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking
4. Impact
Impact

• Measurement
  – We analyzed 61 popular Mac and iOS apps working with ZeroConf
  – 88.5% are vulnerable to man-in-the-middle or impersonation attacks

<table>
<thead>
<tr>
<th>ZeroConf Channels</th>
<th>Vulnerable/Sampled</th>
<th>Sensitive Information Leaked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonjour</td>
<td>18/22</td>
<td>files, directories and clipboard synced, documents printed, instant message</td>
</tr>
<tr>
<td>MC</td>
<td>24/24</td>
<td>files and photos transferred, instant message</td>
</tr>
<tr>
<td>BLE</td>
<td>10/13</td>
<td>User name and password for OS X</td>
</tr>
<tr>
<td>Customized protocols</td>
<td>2/2</td>
<td>remote keyboard input and files transferred</td>
</tr>
</tbody>
</table>
1. ZeroConf Concept
2. ZeroConf How
3. ZeroConf Breaking
4. Impact
5. Protecting ZeroConf
Protecting ZeroConf

• Problem: link a human to her certificate is complicated
• Speaking out Your Certificate (SPYC)
  – Voice biometrics ties certificate to identity
Speaking Out Your Certificate

Hash $h \rightarrow nk$ most significant bits

Partition to $k$ n-bit segments

$\Delta_1 || \Delta_2 || \ldots || \Delta_k \rightarrow <w_1, w_2, \ldots, w_k>$

Words list linking to the certificate
Protecting ZeroConf

• Challenge: link a human to her certificate
• Speaking out Your Certificate (SPYC)
  – Voice biometrics ties certificate to identity
  – Human Subject Study: convenient and effective
Conclusion

• Apple’s ZeroConf techniques are not secure as expected
  – The usability-oriented design affects security
• Addressing such security risks is nontrivial
  – Challenge in binding a human to her certificate
• Our Defense: SPYC
  – Voice biometrics ties certificate to identity