HITBSecConf

Ghost Tunnel

Covert Data Exfiltration Channel to Circumvent Air Gapping

Hongjian CAO, Kunzhe CHAI, Jun LI

PegasusTeam, 360 Security Technology

April 12, 2018

Who We Are

360 Security Technology is a leading Internet security company in Asia. Our core products are anti-virus security software for PC and cellphones.

PegasusTeam was founded in 2015. we focus on the wireless security and wireless penetration testing.

Agenda

- Introduction
- Previous research on Air-Gapped attack
- Ghost Tunnel Introduction
- Ghost Tunnel implementation
- demo

Introduction

- Air-Gapping
- Attack events

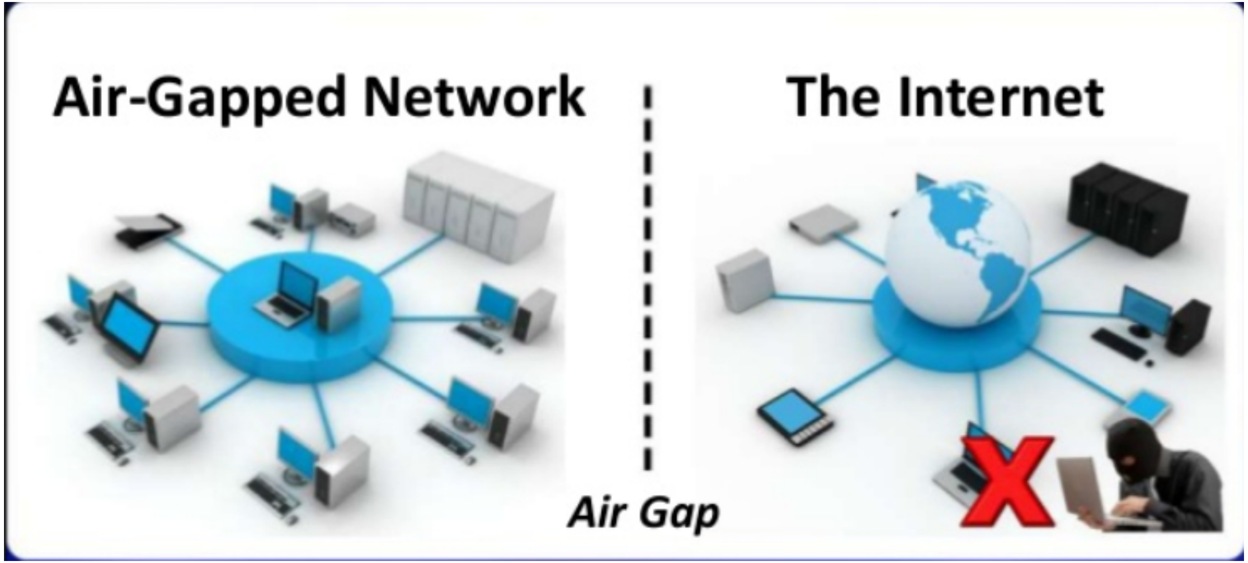
Air Gapping

- 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

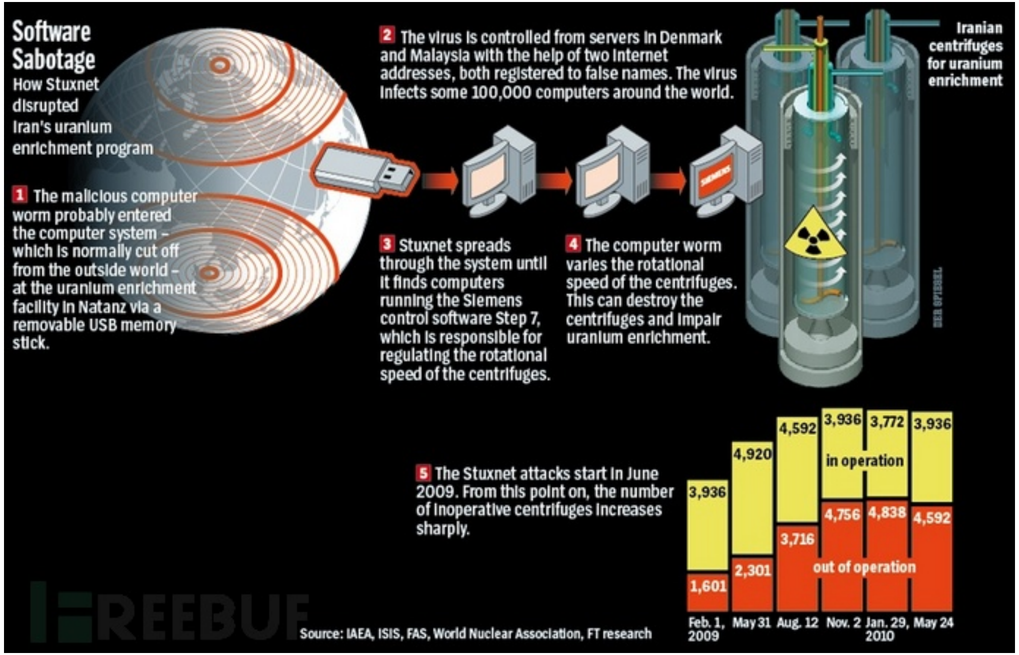


Nothing Is Impossible

- 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



Software Sabotage

How Stuxnet disrupted Iran's uranium enrichment program

1 The malicious computer worm probably entered the computer system – which is normally cut off from the outside world – at the uranium enrichment facility in Natanz via a removable USB memory stick.

2 The virus is controlled from servers in Denmark and Malaysia with the help of two Internet addresses, both registered to false names. The virus infects some 100,000 computers around the world.

3 Stuxnet spreads through the system until it finds computers running the Siemens control software Step 7, which is responsible for regulating the rotational speed of the centrifuges.

4 The computer worm varies the rotational speed of the centrifuges. This can destroy the centrifuges and impair uranium enrichment.

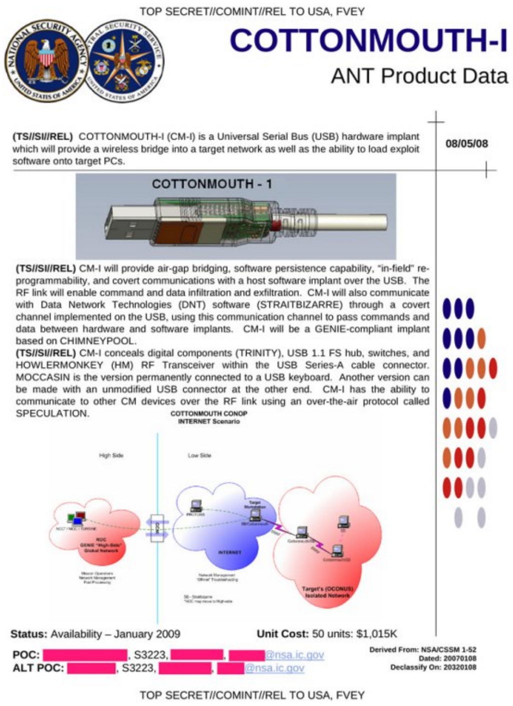
5 The Stuxnet attacks start in June 2009. From this point on, the number of inoperative centrifuges increases sharply.

Iranian centrifuges for uranium enrichment

Source: IAEA, ISIS, FAS, World Nuclear Association, FT research

NSA Leaks (2013)

- COTTONMOUTH-I
  - A USB hardware implant
  - Air-Gap bridging
  - Extracting data from targeted systems via RF signals



TOP SECRET//COMINT//REL TO USA, FVEY

COTTONMOUTH-I

ANT Product Data

08/05/08

(TS//SI//REL) COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

COTTONMOUTH - 1

(TS//SI//REL) CM-I will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-I will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-I will be a GENIE-compliant implant based on CHIMNEYPOOL.

(TS//SI//REL) CM-I conceals digital components (TRINITY), USB 1.1 FS hub, switches, and HOWLERMONKEY (HM) RF Transceiver within the USB Series-A cable connector. MOCCASIN is the version permanently connected to a USB keyboard. Another version can be made with an unmodified USB connector at the other end. CM-I has the ability to communicate to other CM devices over the RF link using an over-the-air protocol called SPECULATION.

COTTONMOUTH CONOP
INTERNET Scenario

Status: Availability – January 2009

Unit Cost: 50 units: $1,015K

POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

Previous research on Air-Gapped attacks

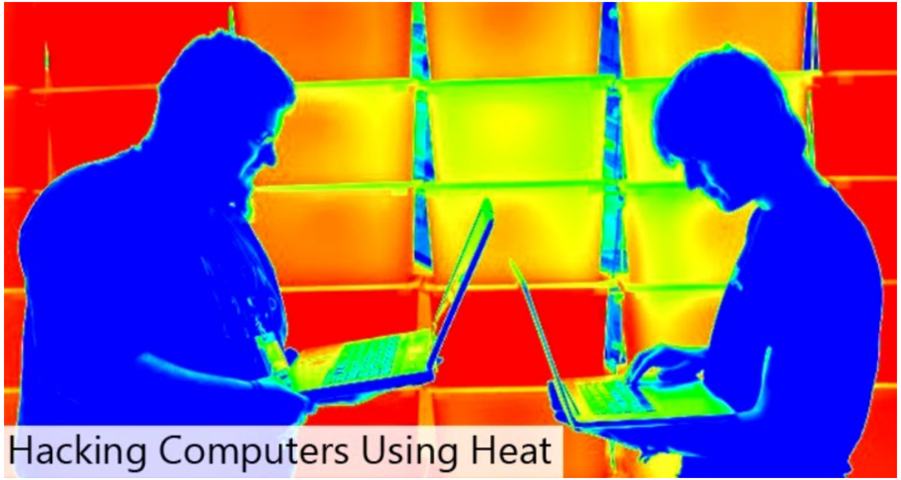
Previous research - 1

- 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

url: https://www.blackhat.com/eu-17/briefings.html#exfiltrating-reconnaissance-data-from-air-gapped-ics-scada-networks

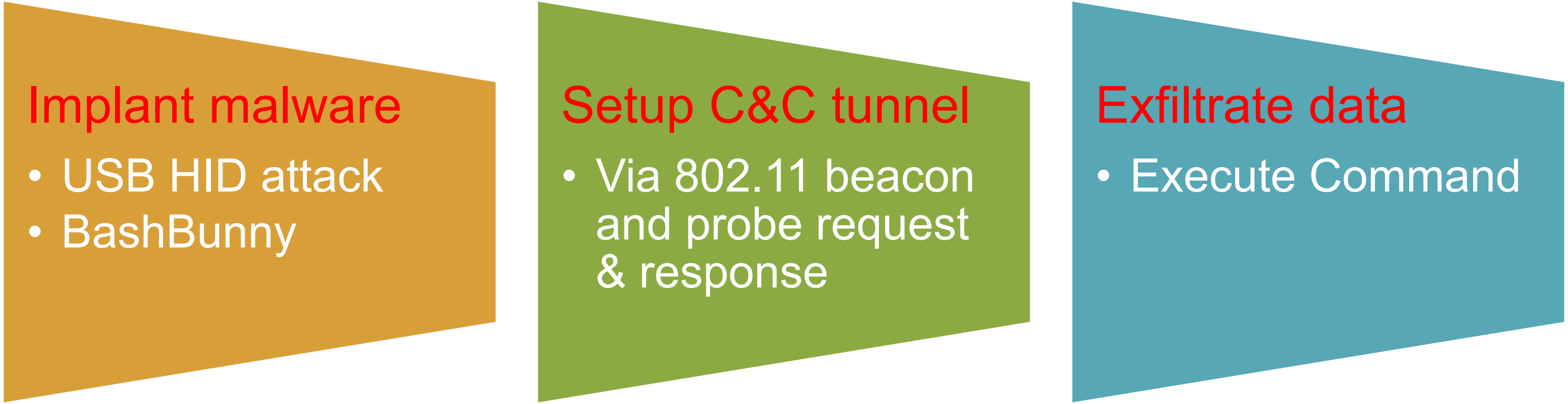
Ghost Tunnel

A Covert Data Exfiltration Channel Using WiFi

Air-gapped Attack

- Implant
  - Malicious software/hardware
- A covert communication channel
  - Any medium that can carry data is possible

Ghost Tunnel



Implant malware

- USB HID attack
- BashBunny

Setup C&C tunnel

- Via 802.11 beacon and probe request & response

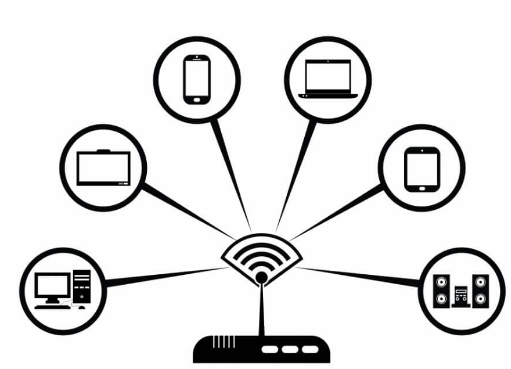
Exfiltrate data

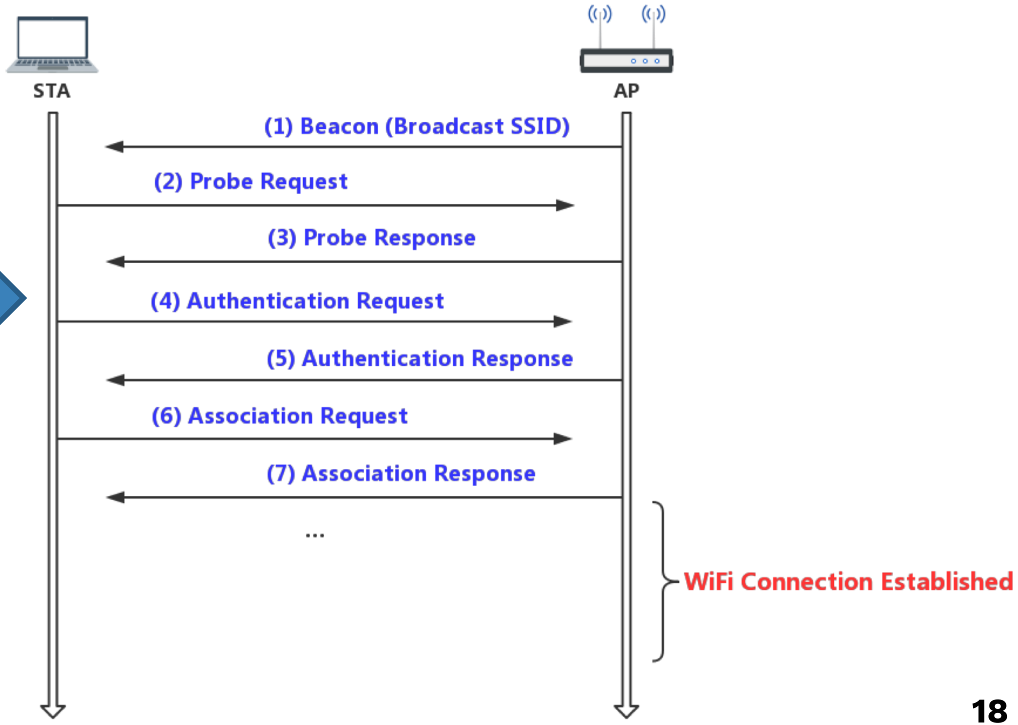
- Execute Command

Ghost Tunnel

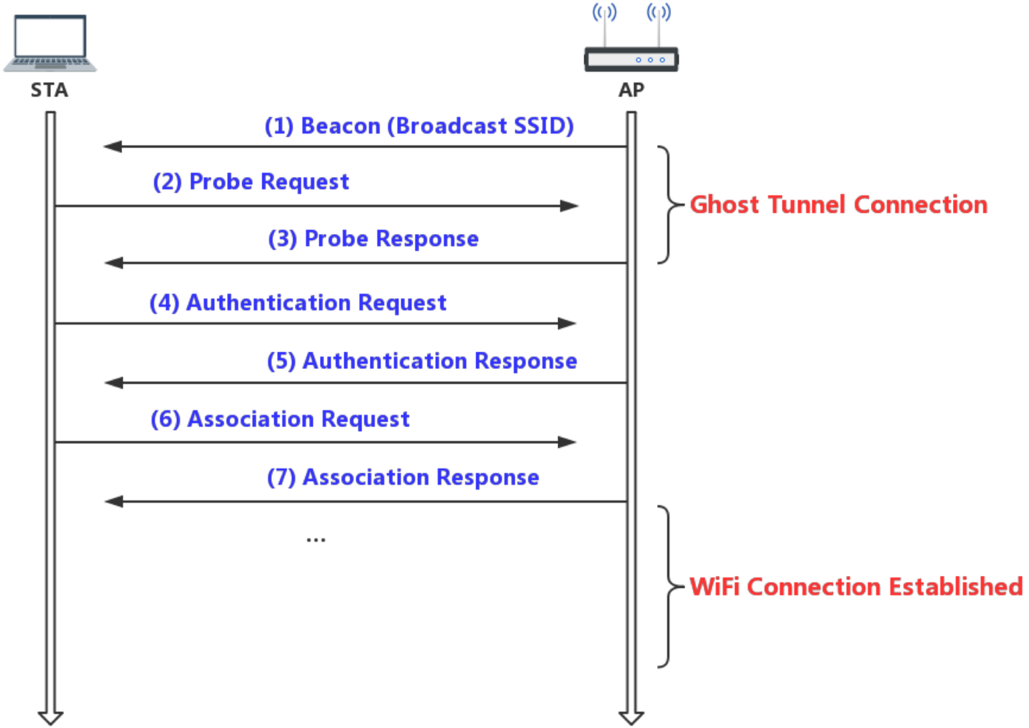
- Can bypass firewalls
- Cross-Platform support
- Allow up to 256 clients
- Effective range up to 50 meters

The Usual Wifi Connection Process

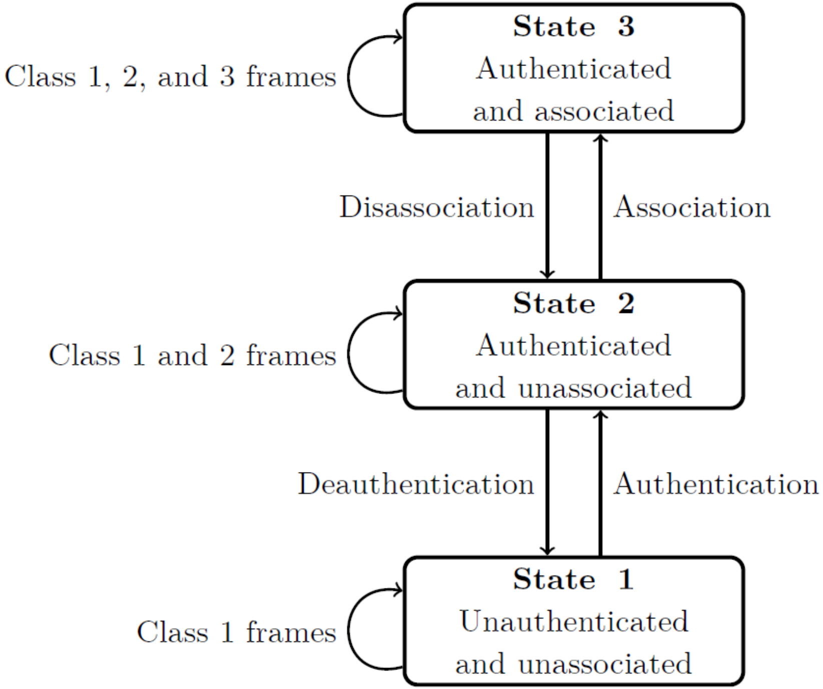



STA
AP
(1) Beacon (Broadcast SSID)
(2) Probe Request
(3) Probe Response
(4) Authentication Request
(5) Authentication Response
(6) Association Request
(7) Association Response
...
WiFi Connection Established

Ghost Tunnel – No WiFi Connection

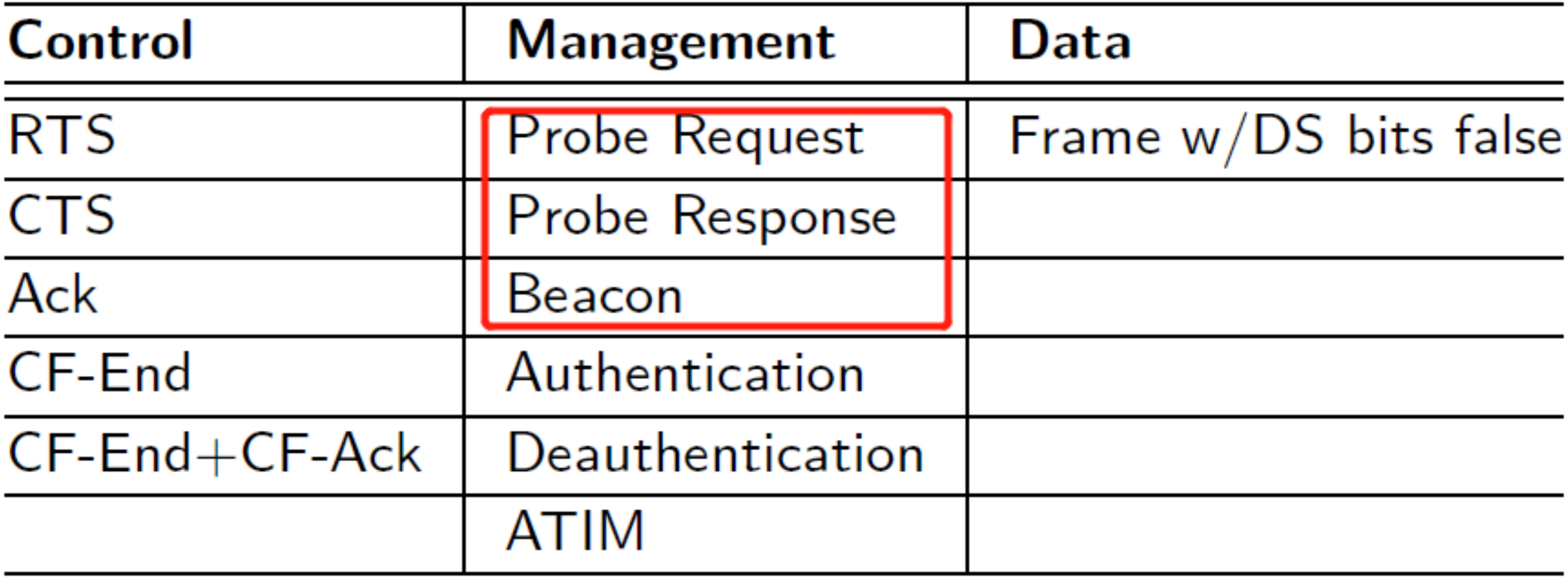


802.11 State



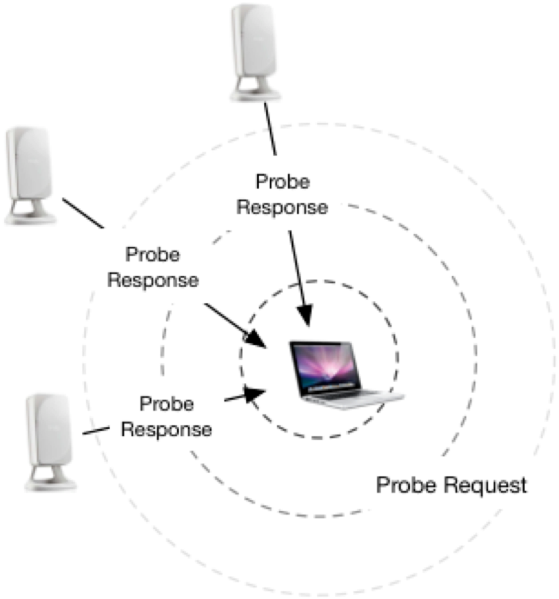
802.11 State Diagram

Class 1 Frames

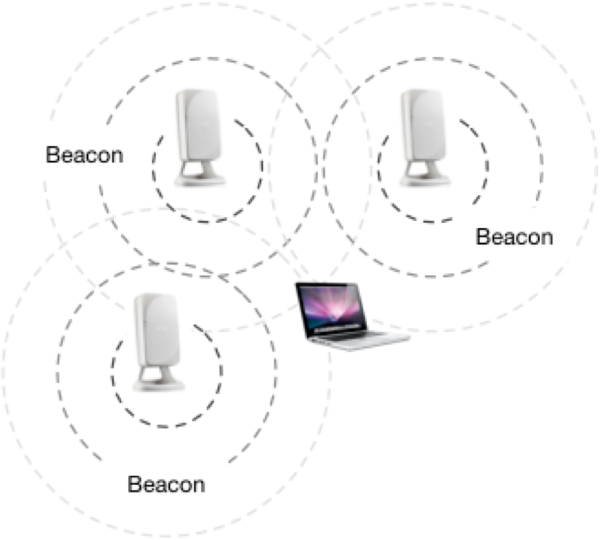


Control	Management	Data
RTS	Probe Request	Frame w/DS bits false
CTS	Probe Response	
Ack	Beacon	
CF-End	Authentication	
CF-End+CF-Ack	Deauthentication	
	ATIM	

Scanning for Wifi Networks



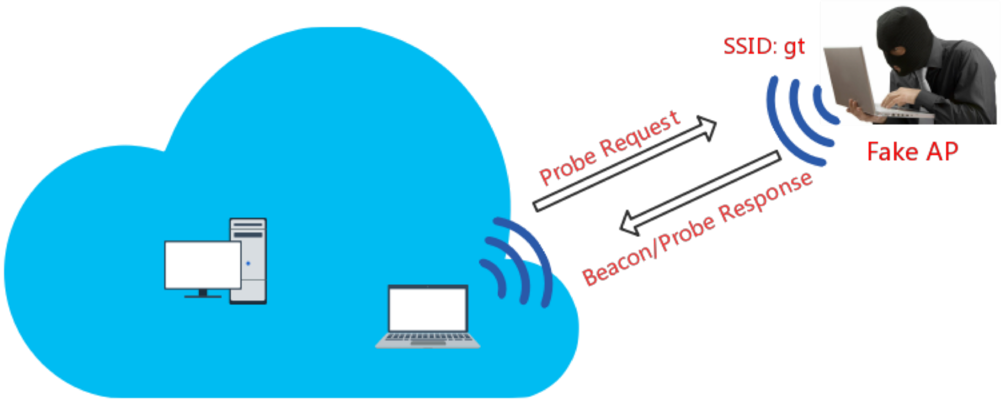
Active Scan



Passive Scan

Ghost Tunnel – No WiFi Connection

- A covert WiFi channel using Beacon, Probe Request, Probe Response
- A special SSID as the identifier



Ghost Tunnel Implementation

802.11 Frame

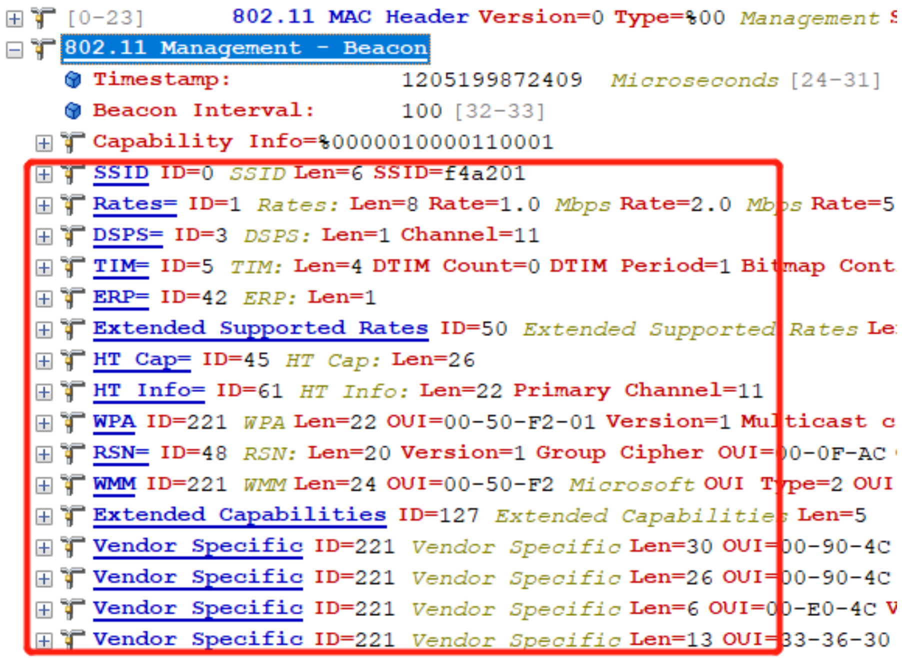
- Control frame
- Management frame
- Data frame

Octets: 2	2	6	0 or 6	0 or 6	0 or 2	0 or 6	0 or 2	0 or 4	variable	4
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Frame Body	FCS

Frame header

802.11 Management Frame Body

- Management Frame Body
  - Fields
  - Information Elements



```
[0-23]        802.11 MAC Header Version=0 Type=%00 Management S
802.11 Management - Beacon
  Timestamp:              1205199872409  Microseconds [24-31]
  Beacon Interval:        100 [32-33]
  Capability Info=%0000010000110001
  SSID ID=0 SSID Len=6 SSID=f4a201
  Rates= ID=1 Rates: Len=8 Rate=1.0 Mbps Rate=2.0 Mbps Rate=5
  DSPS= ID=3 DSPS: Len=1 Channel=11
  TIM= ID=5 TIM: Len=4 DTIM Count=0 DTIM Period=1 Bitmap Cont
  ERP= ID=42 ERP: Len=1
  Extended Supported Rates ID=50 Extended Supported Rates Le
  HT Cap= ID=45 HT Cap: Len=26
  HT Info= ID=61 HT Info: Len=22 Primary Channel=11
  WPA ID=221 WPA Len=22 OUI=00-50-F2-01 Version=1 Multicast c
  RSN= ID=48 RSN: Len=20 Version=1 Group Cipher OUI=00-0F-AC
  WMM ID=221 WMM Len=24 OUI=00-50-F2 Microsoft OUI Type=2 OUI
  Extended Capabilities ID=127 Extended Capabilities Len=5
  Vendor Specific ID=221 Vendor Specific Len=30 OUI=00-90-4C
  Vendor Specific ID=221 Vendor Specific Len=26 OUI=00-90-4C
  Vendor Specific ID=221 Vendor Specific Len=6 OUI=00-E0-4C V
  Vendor Specific ID=221 Vendor Specific Len=13 OUI=33-36-30
```

The components of Information Element

- Element ID: 1 Byte
- Length: 1 Byte
- Information: 0-255 Bytes
  - SSID
  - Vendor Specific

Octets:	1	1	variable
	Element ID	Length	Information(payload)

Element Format

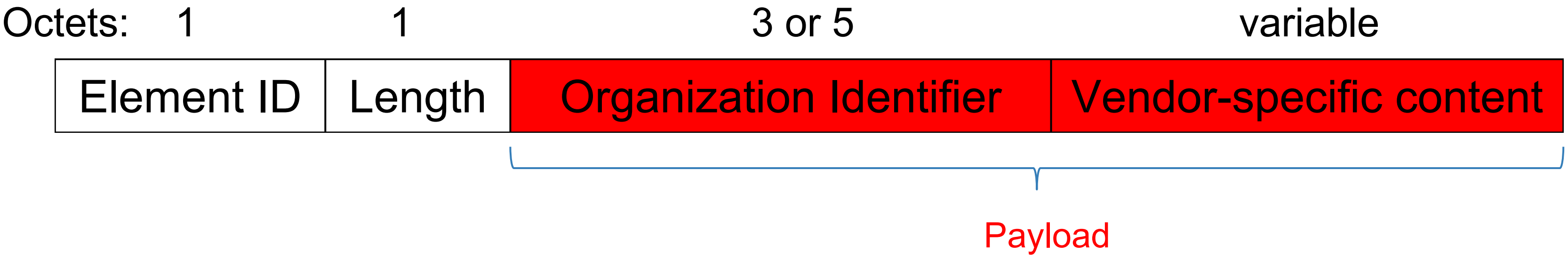
SSID Element

- Identity of an ESS or IBSS
- SSID length 0-32 Bytes

Octets:	1	1	0-32
	Element ID	Length	SSID(Payload)

Vendor Specific Element

- ID = 221
- Organization Identifier
- Vendor-Specific content



Octets:	1	1	3 or 5	variable
	Element ID	Length	Organization Identifier	Vendor-specific content

Key Problem

- How to send and receive 802.11 data frames through local wireless network interface in user space ?

- Wireless network interface mode
  - Master (Acting as an AP)
  - Managed (Station)
  - Monitor (Monitor all traffic)
  - …

Through Operating System WiFi API

- Windows
  - Native Wifi API
- Mac OSX
  - coreWLAN
- Linux
  - nl80211 & libnl

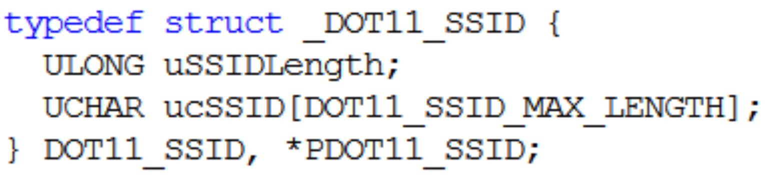
Windows Client: Send And Receive

```
DWORD WINAPI WlanScan(
            _In_  HANDLE   hClientHandle,
            _In_  const GUID   *pInterfaceGuid,
            _In_opt_ const PDOT11_SSID    pDot11Ssid,
            _In_opt_ const PWLAN_RAW_DATA pIeData,
            _Reserved_  PVOID pReserved );
```

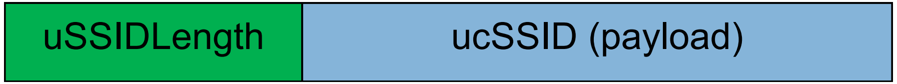
- scan for available wireless networks
  - pDot11Ssid, specifies the SSID of the network to be scanned
  - pIeData != NULL, send probe request
  - pIeData == NULL, not send probe request

Packet payload Format

- DOT11_SSID
  - Contains the SSID
  - The maximum length is 32

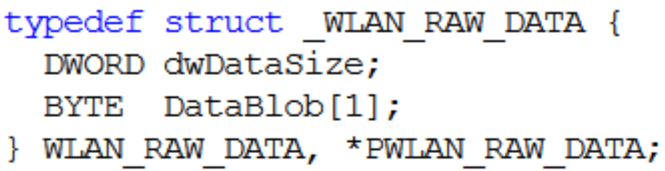


```
typedef struct _DOT11_SSID {
  ULONG uSSIDLength;
  UCHAR ucSSID[DOT11_SSID_MAX_LENGTH];
} DOT11_SSID, *PDOT11_SSID;
```

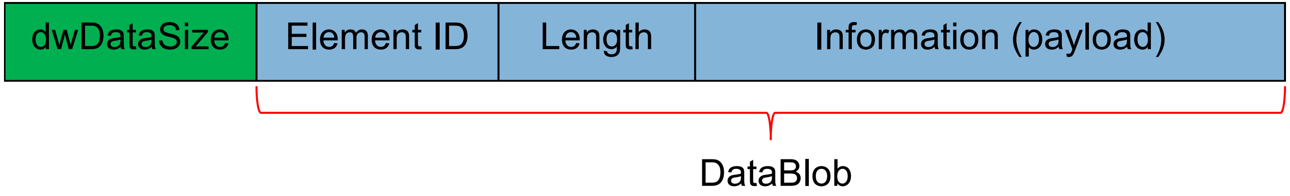


uSSIDLength	ucSSID (payload)

- WLAN_RAW_DATA
  - Contains the elements data
  - Not exceed 240 bytes



```
typedef struct _WLAN_RAW_DATA {
  DWORD dwDataSize;
  BYTE  DataBlob[1];
} WLAN_RAW_DATA, *PWLAN_RAW_DATA;
```



dwDataSize	Element ID	Length	Information (payload)

DataBlob

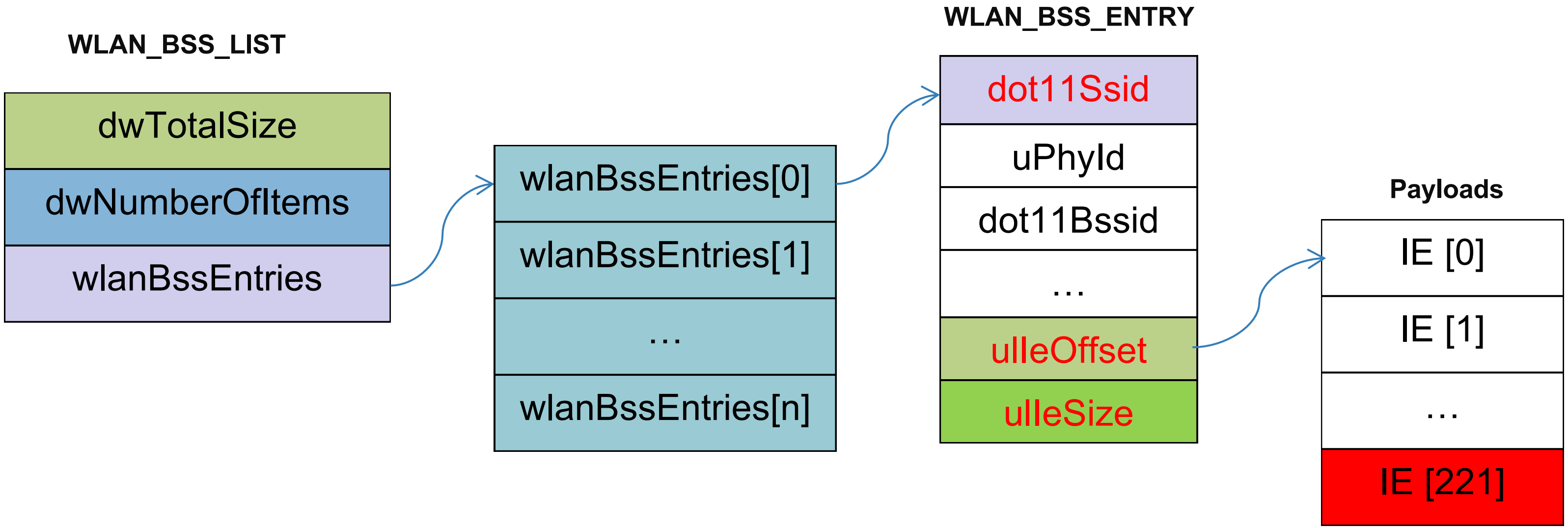
Windows Client : Receive

```
DWORD WINAPI WlanGetNetworkBssList(
          _In_   HANDLE                    hClientHandle,
          _In_   const GUID           *pInterfaceGuid,
          const PDOT11_SSID                pDot11Ssid,
          _In_   DOT11_BSS_TYPE            dot11BssType,
          _In_   BOOL                      bSecurityEnabled,
          _Reserved_   PVOID       pReserved,
          _Out_   PWLAN_BSS_LIST *ppWlanBssList );
```

- Retrieve available wireless networks list
- ppWlanBssList
    - Receive the returned list of of BSS entries

Windows Client : Receive

- WLAN_BSS_LIST
  - An array of WLAN_BSS_ENTRY structures that contains information about a network



Mac Client : Send

- CWInterface
  - func scanForNetworks(withSSID: Data?)

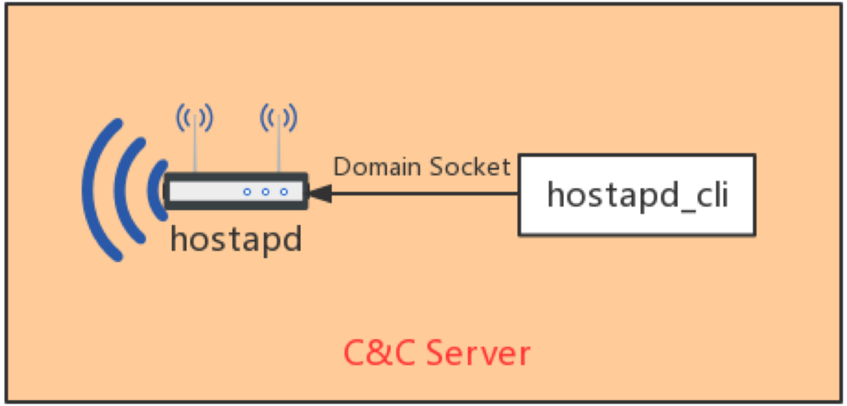
Mac Client : Receive

- CWInterface
  - func scanForNetworks(withSSID: Data?)
  - func cachedScanResults() -> Set<CWNetwork>?

- CWNetwork
  - informationElementData: Data?

C&C Server: Send And Receive

- Modified hostapd and hostapd_cli
- USB WiFi card



Demo

Ghost Tunnel

360PegasusTeam

Thanks!

Any questions?