The .11 Veil, Covert and Camouflage /*Invisible WIFI Revealed*/

Who are we…

- Amrita C. Iyer
- Rushikesh D. Nandedkar

Agenda

- **Introduction and some details**
- **Covert Communication**
- **Elt Euphoria**
- **The Patch Peloton**
- **Things we learned (|| learning ? ☺)**
- **Potential approaches**
- **Conclusions**
- **Acknowledgements**

Super cluttered, isn’t it? Indeed yes !!!

Krghveiutyrwilwqekhwnilktbuhlon
mjgasvl ;adjsnvlkgfjb kjfdgtbnzs
kgvashebvzkjeargkjbgdjtlkynjrdsbl
ktrhwvszsakjrbgasdbljljenlidsselkh
wlnksdlksafnkjgsanvldelkd;lobesiv
hwerpqoipn09v2i65rpe9i6umn;otirjnt
h oilesdbqwoinweilueiomyujnrspo
potujuyrvgeuy4eiy54wqiy4e49puh4jo
u4eow4u34qVOPEUNW[OMBEYUNK
REMPYUE0BMWRNWOVYRWOIEBIEI
YWIOTYENTVWQIYEBIWOYU98RMY
URE9YU4SROILAURDTVBENIHYN3BE

© J. Arnold and CELF

And Sorted …… !

DON'T

TRUST PAYLOADS

TRUST: ~~ACTIONS~~ BEACONS

[The story (myth???)] || !(myth???)

Did I just say grey ?

- There are always untold pieces in theory story.
- Something that they try to prevent disclosing.
- There are always “otherwise” usages of things.
- And so is wireless, the holy .11 (Edit, IEEE 802.11).

The characters

- Access point (AP)
- Host

Effects Utilities Window Help
WE NEED TO GO DEEPER

Back to the basics

INTRODUCTION & PROLOG

./../

- What .11 is blamed for?
- Victims
- .11 modes

What .11 is blamed for?

- Do we need a proof to call bug a bug?
- Well ok ↴
  - A hole in the network perimeter (open wireless networks, wep, bad configs).
  - Loose link in client’s security:
    - Offensive rogue access points
    - Eavesdropping in socially dense areas
    - Connectivity messups

Victims

Courtesy to the omnipresence and ease of access of wireless:

- Mobile phones
- Cameras
- Printers
- Gaming consoles
- Laptops, desktops …. …. …. ….

More and more places to be equipped with wi-fi.

All in all, many victims ………………..… awaiting exploitation ☺ !

.11 modes

I. Managed: acts as a station
II. AdHoc: acts as an AdHoc station
III. Master: acts as an access point
IV. Monitor (RFMON): shows everything seen by radio. (synonymous to promiscuous mode in .3)

Covert Communication

By book…

- In computer security, a **covert** channel is a type of computer security attack that creates a capability to transfer information objects between processes that are not supposed to be allowed to **communicate** by the computer security policy.

- There have always been ways to smuggle the data using various layers in the ISO OSI model.
- We have been focusing on some of the aspects in data link layer.
- And that too specifically on beacons and probes.

.11 Frame Types

- Management frames
- Control frames
- Data frames

Management Frames

- Association request
- Association response
- Re-association request
- Re-association response
- **“Probe”** request
- **“Probe”** response
- **“Beacon”** frame

Control frames

- Request to send
- Clear to send
- Ack
- PS poll …

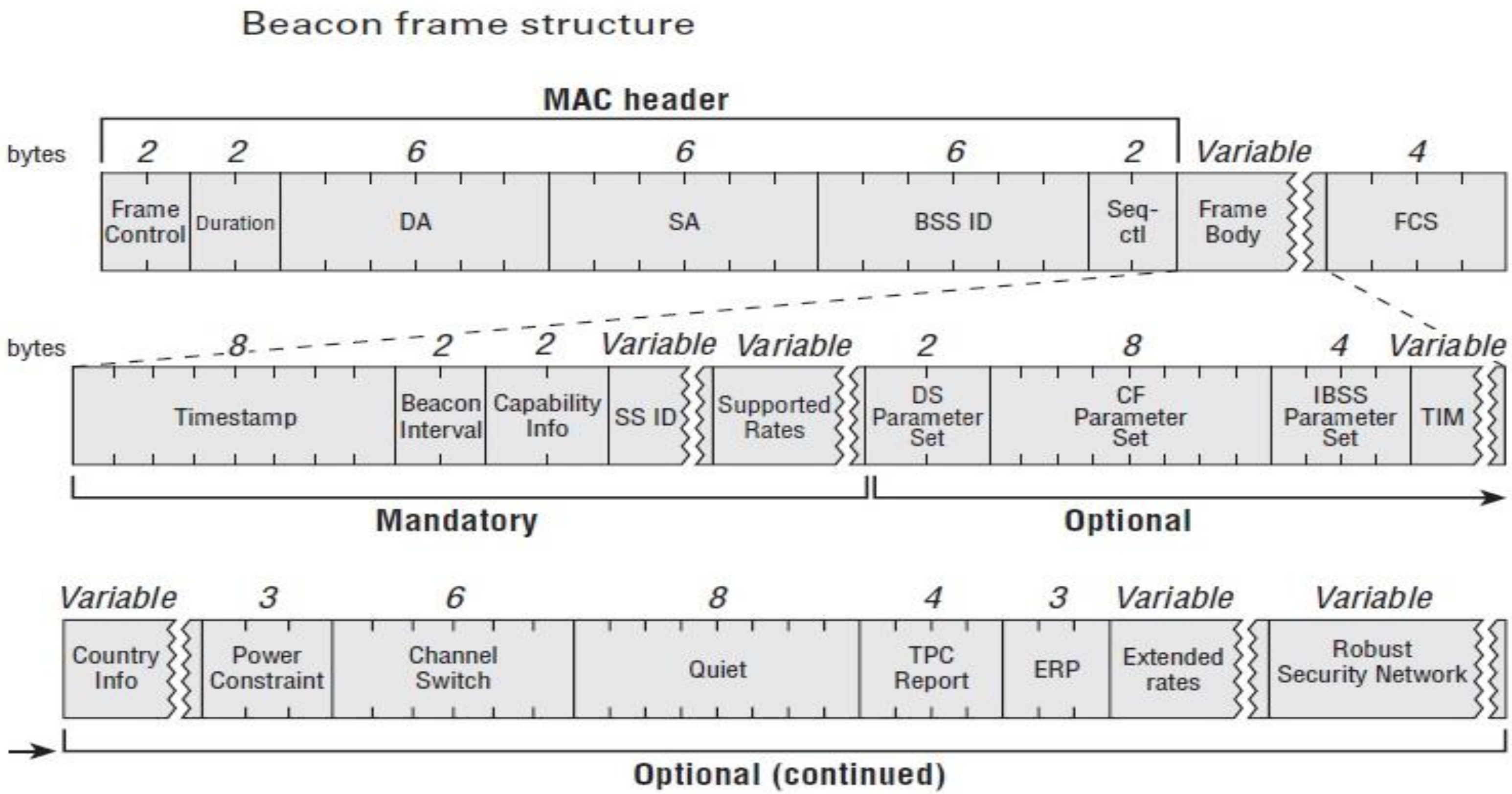
Data Frames

- A-MSDU
- Variants of MPDU …

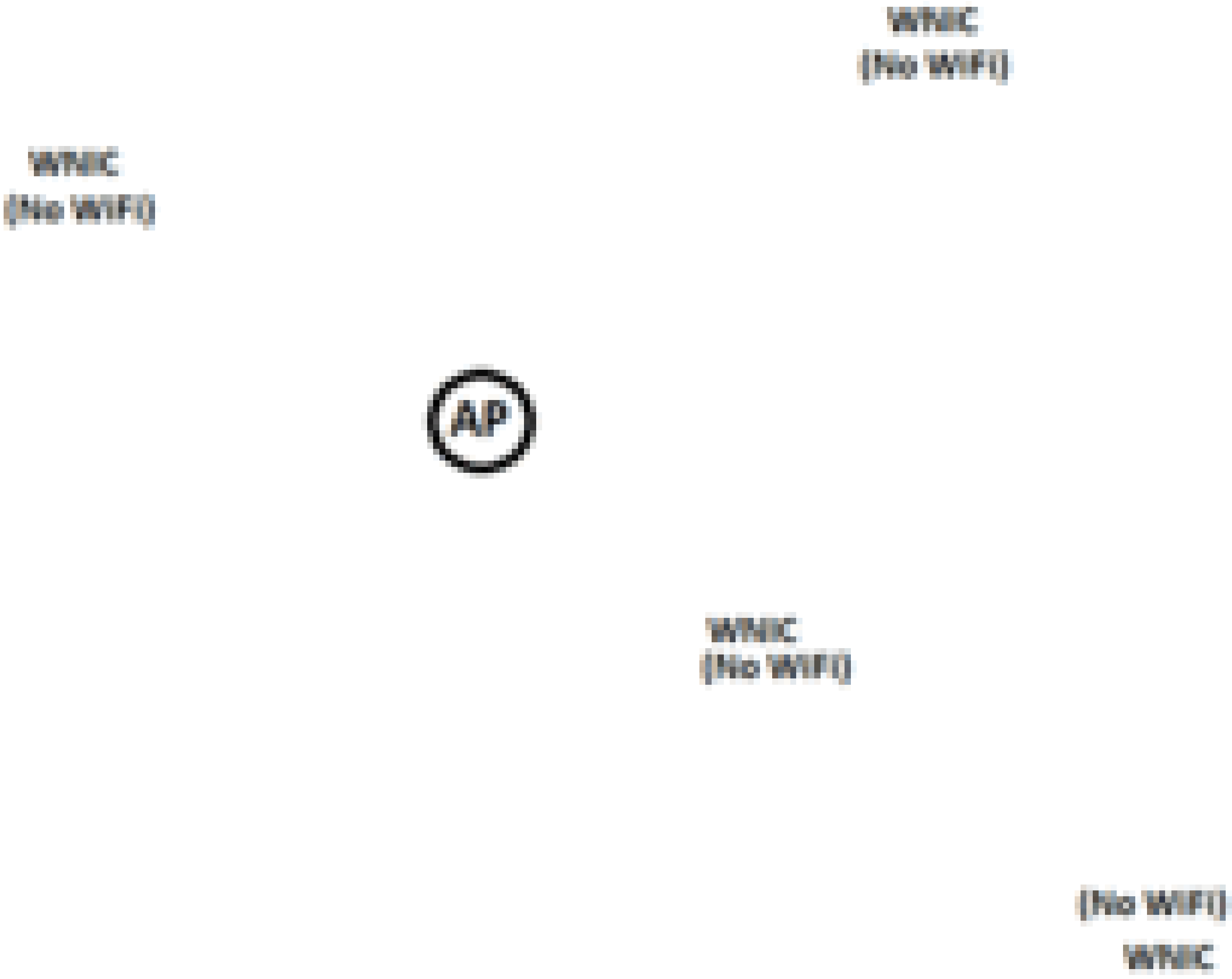
Elt Euphoria

- Elt is nothing but information element (part of wireless frames).
- Beacon frame is essential element in the wireless networks.
- Beacon frame populates air with a rate of around one frame per 100 milliseconds.
- They are abundantly available.
- They are broadcasts.
- Requires no authentication and/or association with access points to listen to them.

Beacon frame



Source: https://mrncciew.files.wordpress.com/2014/10/cwap-mgmt-beacon-01.png


WNIC
(No WIFI)
WNIC
(No WIFI)
AP
WNIC
(No WIFI)
(No WIFI)
WNIC

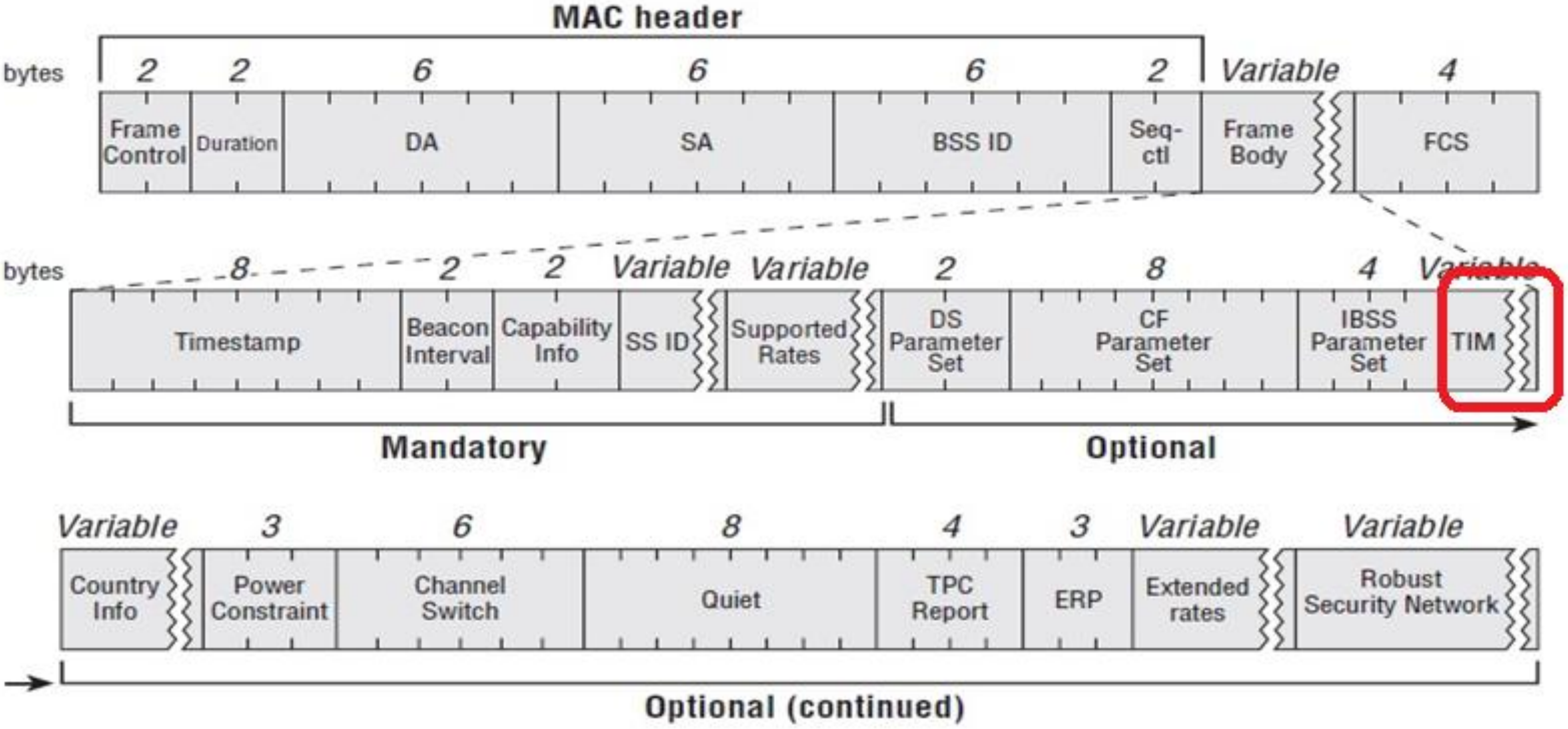
- There is a lot of information stuffed inside the wireless frames (in our context, beacon frame).
- So how to harness the true power of these frames.
- Edit the fields which have better lengths in order to ship data.
- Interesting elements: SSID, DSset, TIM, Rates, ESRates, TPC Requests/responses, country etc.

Why Beacon/Probe Frames?

- Beacon/Probe frames does not require auth and association to air themselves.
- Being broadcast, so no need to zero down on host selection. Reduces the pain a little bit!
- Presence of theses frames in multitude in local wireless periphery is common phenomenon, hence escapes suspicious eyes initially.
- Again the multitude will always facilitate the larger chunk of data to ship

- Outbreak of malware? very much a possibility!
- Some fields allows pushing more than 250 bytes of data in a single frame.
- 250 bytes are quite enough for malicious payload.

Beacon frame structure



MAC header

bytes	2	2	6	6	6	2	Variable	4
	Frame Control	Duration	DA	SA	BSS ID	Seq-ctl	Frame Body	FCS

bytes	8	2	2	Variable	Variable	2	8	4	Variable
	Timestamp	Beacon Interval	Capability Info	SS ID	Supported Rates	DS Parameter Set	CF Parameter Set	IBSS Parameter Set	TIM

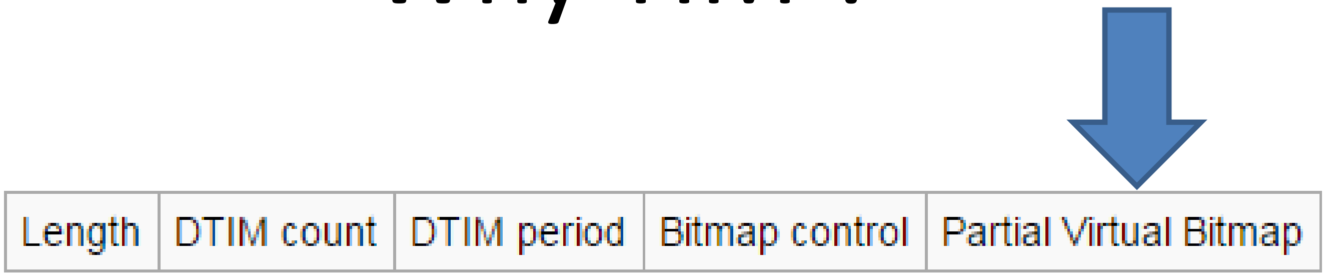
Mandatory

Optional

Variable	3	6	8	4	3	Variable	Variable
Country Info	Power Constraint	Channel Switch	Quiet	TPC Report	ERP	Extended rates	Robust Security Network

Optional (continued)

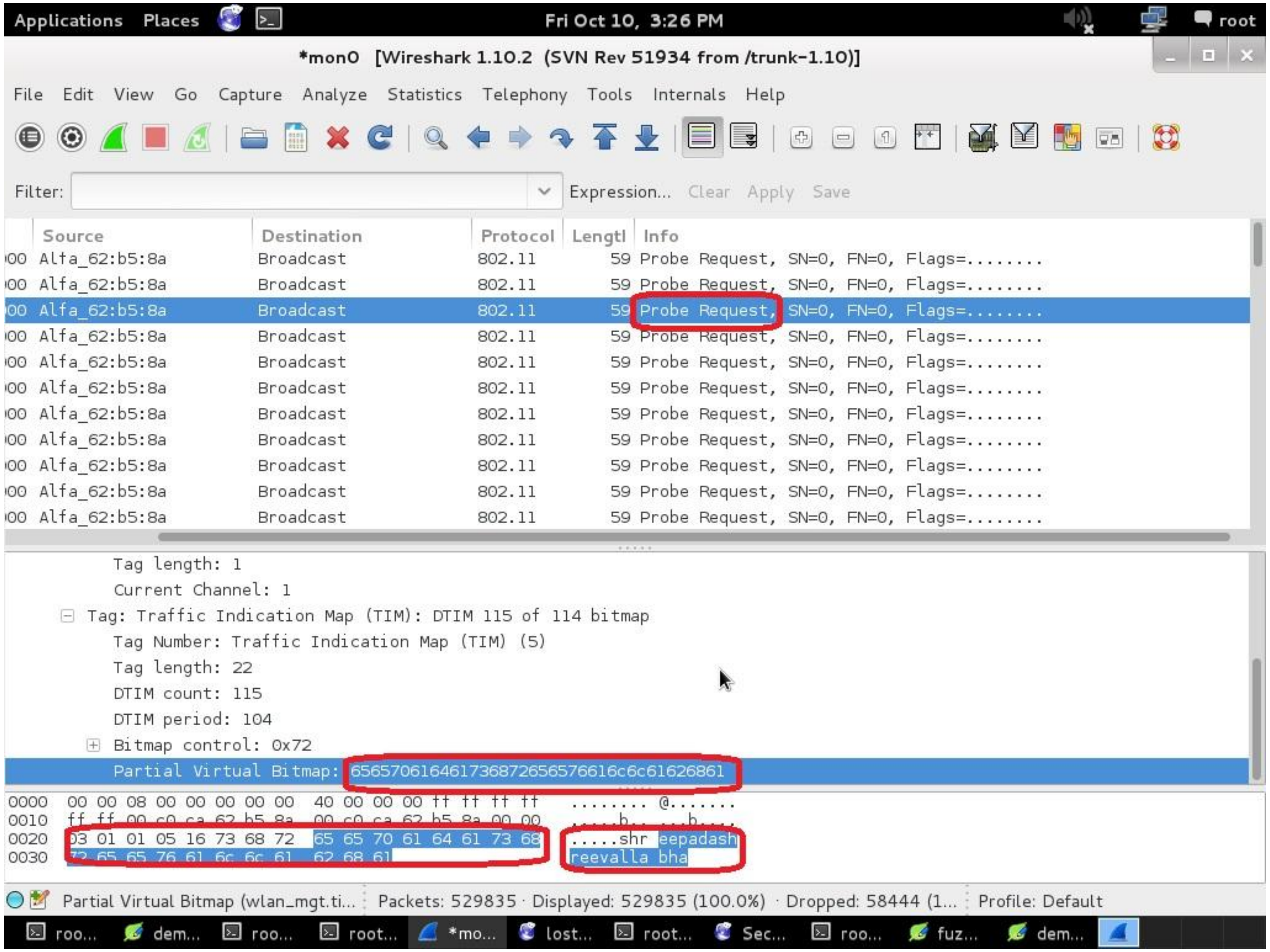
Why TIM ?



1- 251 bytes

Length	DTIM count	DTIM period	Bitmap control	Partial Virtual Bitmap

- TIM allows shipping data of around 250 bytes in the Partial Virtual Bitmap field.
- Essentially, it was easy to fabricate the frame in scapy with this information element.



```
   Source           Destination   Protocol  Lengtl  Info
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
00 Alfa_62:b5:8a    Broadcast     802.11        59  Probe Request, SN=0, FN=0, Flags=........
```

```
        Tag length: 1
        Current Channel: 1
    ⊟ Tag: Traffic Indication Map (TIM): DTIM 115 of 114 bitmap
        Tag Number: Traffic Indication Map (TIM) (5)
        Tag length: 22
        DTIM count: 115
        DTIM period: 104
      ⊞ Bitmap control: 0x72
        Partial Virtual Bitmap: 65657061646173687265657661 6c6c61626861
```

```
0000  00 00 08 00 00 00 00 00  40 00 00 00 ff ff ff ff   ........ @.......
0010  ff ff 00 c0 ca 62 b5 8a  00 c0 ca 62 b5 8a 00 00   .....b.. ...b....
0020  03 01 01 05 16 73 68 72  65 65 70 61 64 61 73 68   .....shr eepadash
0030  72 65 65 76 61 6c 6c 61  62 68 61                  reevalla bha
```

Partial Virtual Bitmap (wlan_mgt.ti... Packets: 529835 · Displayed: 529835 (100.0%) · Dropped: 58444 (1... Profile: Default

Raw scapy script

- #!/usr/bin/env python
- from scapy.all import *
- srcmac = "00:23:66:E2:F3:2E:3A"
- dstmac = "ff:ff:ff:ff:ff:ff"
- ssid   = Dot11Elt(ID="SSID",info=“AAAAAAAA”)
- #tim   = Dot11Elt(ID="TIM",info=“bruconbruconbruconbrucon”)
- pkt = RadioTap()/Dot11(type=0,subtype=4,addr1=dst)/Dot11Elt(ID=5,len=200,info=bruconbruconbruconbrucon)
- [Note: still facing issue with this script]

Issues:

- Deep packet inspection firewalls may prove of trouble here.
- Reordering the data at receiver end could be an issue, should sequencing is not taken care of before shoving in the data.
- No retrieval of lost frames so far.
- Scapy doesn't support Beacon Injection swiftly still.

A minute diversion to the Elt Euphoria

- **ACK** frames or **RESPONSE** frames are of significance to reply to certain communication initiated by the remote host earlier.
- The trust is already in place between two hosts.
- The responses or acknowledgements sent by unsolicited user will receive little low priority of inspection as it has been assumed that such responses are bound to come from a legit source on peripheral devices.

- Adding this approach with the Elt Euphoria will give solution to the sequencing issue.
- The response traffic is always made more intelligent as they are capable of assigning sequence and discipline the traffic at receiver end.
- The parameters which could come handy are, Frame Control, Frame Control Sequence, More Data, More Fragments, Sequence Numbers, BSSID, ESSID and essentially **"Source Address"** etc.

Recipe

- 1.1 Encode the data and ship it over the ACK.
- 1.2 Use the ID parameter to encode.
- 1.3 Share this magic parameter with the receiver.
- 1.4 Run the partial stealth mode on legit ACK.

- This may lead to Ad-Hoc network scenario.
- Resulting in more autonomy and more control over the data.

Issues

- Anomaly based detection is possible.
- The lost frames issue is still unattended, not much help from Retry field.

The Patch Peloton

- The driver patching is one of the most efficient way of achieving invisibility in the air.
- This approach fairly mitigates the issues we have confronted in the previous approaches.
- Having this said, it is truly covert conduit setup for securing the communication over the air.

The test case

- Prepare two hosts (**unpatched drivers**, **linux machines**, Windows machines will do as well) for scanning/stumbling purpose.
- Raise an access point on one linux machine by tuning into **MASTER mode** with having the **patched drivers**.
- The machines with **unpatched drivers** will **not** be **able** to see the **"Engineered Traffic."**
- The machines with **patched drivers** will be **able** to communicate with other devices having same **patched protocol stack**.

The deductions from this approach are:

- Engineered beacon frames from Access Points with patched protocol stack were not read by the devices having unpatched protocol stack.
- Neither of probes injected by devices with patched protocol stack were read by the devices with unpatched version of protocol stack.
- Sniffers gave little variation in the dump of traffic. In some cases devices with unpatched protocol stack were not able to sniff engineered traffic at all. And some dumps gave a garbled traffic.

Advantages

- In house solution for mitigating majority of attacks on Wireless Infrastructure.
- Partial occurrence of **Event Horizon** in Wireless Networks is very much achievable using this approach.
- Requires no great deal of changes in the operating environments other than patched drivers
- Low Cost Low Effort solution.

Things we learned
(|| learning ?)

- Issues with scapy, as far as beacon frame injection is concerned.
- Building patches takes a lot of input from various sources.
- It grew more complicated in 4.* series of linux kernels, to build a patch.

Potential Approaches

- Lot of information elements are yet to tested.
- We recently found TPC request/responses are capable of doing similar traffic.
- We have explored only version field in the driver patching.
- PS-Poll frame is also an interesting carrier, yet we could not work the traffic so far.

Conclusion

- Wireless networks (IEEE 802.11) have a different way of securing as well, by mean of running covert channels.
- The approaches we have proposed are still in development so far which with the help of minute automation can lead to nicer outcomes.

Acknowledgements

- Vivek Ramachandran (Wireless security megaprimer)
- Josh Wright (for scapy scripts)
- And **BruCON 2015 ……..**