EXPERIMENT 05

Aim: Prepare a case study related to above mentioned tools.

Theory:

KISMET:

Kismet application is an open source wireless network analyzer running on Linux, UNIX and Mac OS X. It is not supported by Windows OS. Kismet is a passive sniffer used to detect any wireless 802.11a/b/g protocol complaint networks, even when the network has a non broadcasting hidden SSID (Secure Service set Identifier).

Kismet can discover, log the IP range of any detected wireless network and report its signal and noise levels. It can sniff all management data packets from detected networks. Kismet can be used to locate, troubleshoot and optimize signal strength for access points and clients, as well as detect network intrusions.

KISMET CLIENT AND KISMET SERVER :

The kismet protocol is used by the kismet server and kismet client to control the server and its capture sources. Kismet server is controlled from the kismet.conf files located in the /usr/local/etc directory. The kismet.conf is where most of the kismet server configuration is done. Here the wireless adapter‟ or „Source‟ is configured on the client computer and or , configured to indicate that the drone is of a remote source like kismet_drone, while using a special service port like port 3501. Another kismet protocol is the kismet drone/kismet server protocol used by
the kismet server to communicate with a remote drone. Here configuration changes can be made in the kismet_drone.conf file, by modifying the „Source‟ and „allowed host‟ files to suit the end users network segment and drone type and version.

The drone runs as a daemon; being able to launch at boot time and run in the background responding to network service request/ hardware activities and forwarding the request to other processes. For example, packet requests from available network hosts are processed by the kismet drone and sent to the appropriate local client server port. My MacBook laptop comes with Broadcom BCM4321 wireless card which supports RFMON in its wl driver. The wl driver is not capable of enabling monitor mode. To enable the RFMON capability, there are several options and one of them is the drone method which was adopted in this thesis. The

other option would have been to load an ndis driver using ndiswrapper application into the bcmwl5.sys file found inside the BCM driver. The ndiswrapper is patched with bcmmon.diff, a common binary file. Through a series of compilation and configuration the Broadcom driver is enabled. This method involves tweaking the OS kernel and requires some good knowledge of root sources in the OS core. The drone applications are simple to configure and compile without having to temper so much with the kernel.

NETSTUMBLER

NetStumbler or Network Stumbler is a free downloadable software or tool for windows. It can detect wireless LAN using the IEEE802.11a/b/g WLAN protocol standard. NetStumbler is commonly used for:

- WarDriving
- Verifying network configurations
- Finding locations with poor coverage in a wireless local area network (WLAN)
- Detecting causes of wireless interference
- Detecting unauthorized "Rogue" access points
- Aiming directional antennas for long haul WLAN links
- Can be integrated with GPS for mapping purposes.

Some of its limitations are:

- NetStumbler software does not officially work very well on windows vista or Mac OS.
- It uses active scanning to detect access points by sending out beacon probes requests every second and then recording the responses. This makes it vulnerable for detection in a wireless environment.
- It cannot detect wireless stations, since wireless stations do not respond to active probe requests.
- There are numerous alternative NetStumbler tools available today, all having different functionalities, purpose, strength and weaknesses.

Here are some of them:

- MacStumbler
- iStumbler
- Windows Vista Netsh
- Vistumble
- Inssider

• DISA Wireless Discovery Device (Flying Squirrel)

This thesis work shall only examine NETSH which comes by default in windows vista.

Android OS Security Features:

• App sandbox: The Android platform takes advantage of the Linux user-based protection to identify and isolate app resources. To do this, Android assigns a unique user ID (UID) to each Android app and runs it in its own process. Android uses this UID to set up a kernel level App Sandbox.

• App signing

App signing allows developers to identify the author of the app and to update their app without creating complicated interfaces and permissions. Every app that runs on the Android platform must be signed by the developer.

• Authentication

Android uses the concept of user-authentication-gated cryptographic keys that requires cryptographic key storage and service provider and user authenticators.

On devices with a fingerprint sensor, users can enroll one or more fingerprints and use those fingerprints to unlock the device and perform other tasks. The Gatekeeper subsystem performs device pattern/password authentication in a Trusted Execution Environment (TEE). Android 9 and higher includes Protected Confirmation, which gives users a way to formally confirm critical transactions, such as payments.

• Biometrics

Android 9 and higher includes a BiometricPrompt API that app developers can use to integrate biometric authentication into their apps in a device- and modality-agnostic fashion. Only strong biometrics can integrate with BiometricPrompt.

• Encryption

Once a device is encrypted, all user-created data is automatically encrypted before committing it to disk and all reads automatically decrypt data before returning it to the calling process. Encryption ensures that even if an unauthorized party tries to access the data, they won't be able to read it.

• Verified Boot

Verified Boot strives to ensure all executed code comes from a trusted source (usually device OEMs), rather from an attacker or corruption. It establishes a full chain of trust, starting from a hardware-protected root of trust to the bootloader, boot partition and other verified partitions.

App Security:

• Source code review can detect a broad range of security issues, including those identified in this document. Android strongly encourages both manual and automated source code review.

• Automated testing

Automated testing can help detect a broad range of security issues and should be performed regularly.

• Vulnerability scanning

Vulnerability scanning can help ensure that pre-installed apps are free of known security vulnerabilities. Advanced detection can reduce the time and cost required with addressing these vulnerabilities and preventing risk to users and devices. Scan all pre-installed apps using an industry-recognized app vulnerability scanning tool and address detected vulnerabilities.

• Scan all pre-installed apps using an industry-recognized app vulnerability scanning tool and address detected vulnerabilities.

Potentially Harmful Applications

It is important to ensure that the pre-installed apps on your device aren't Potentially Harmful Applications (PHAs). You are responsible for the behavior of all apps that are included on your devices. Prior to device launch, scan all pre-loaded apps for vulnerabilities.

• Isolating apps and processes

The Android sandboxing model provides extra security around apps and processes when used correctly.

Network Security:

• Securing listening sockets

Use listening sockets with caution. There should generally not be any open listening sockets on devices as these provide a vector for a remote attacker to gain access to the device.

• Disable ADB

Android Debug Bridge (ADB) is a valuable development and debugging tool, but is designed for use in a controlled, secure environment and should not be enabled for general use. Example


Vidyavardhini's College of Engineering and Technology, Vasai
Department of Computer Science & Engineering (Data Science)

Fing network scanner app:
Advantages and Disadvantages

Advantages:

Fing is a free tool that can be downloaded from the internet.

It can detect all devices connected to a network and ensure its health and safety. Fing can accurately display the name of some devices, but for others, it only provides related details.

It can detect the presence of other Wi-Fi networks in the same location, which serves as a warning that an Airbnb could potentially have hidden cameras on another network. Fing has a premium subscription, which unlocks more security features, including the ability to specifically scan for cameras.

Disadvantages:

Fing is not compatible with all wireless cards and may cause instability in Fing itself. It can be detected easily by most intrusion detection systems because it actively probes a network to collect information.

Fing is not supported on older versions of Android.

It does not work with wireless networks that use encryption.

Fing is not open source software.