BRING YOUR OWN DEVICE (BYOD) DETECTION USING PEREGRINE7 CENTRAL MANAGEMENT SYSTEM

Kavyashree J[1], Pavanalaxmi[2]
[1]M.tech 4th sem, VLSI & embedded system, SCEM, Mangalore
[2]Dept of ECE, SCEM, Mangalore

***Abstract* - Peregrine7 is a BYOD (Bring Your Own Device) Security and Access Control Solution. It provides real-time monitoring, mitigation of enterprise risk associated with noncompliant and/or compromised endpoints, and device & app aware granular access control over enterprise resources.**

***Index Terms* - Peregrine7, postgresql**

I. INTRODUCTION

Peregrine Guard provides one of the best enterprise mobility network access control platform and integrates well with the existing authentication products such as MS Active Directory, LDAP and RADIUS and also works well with the existing security and network infrastructure with no changes to the existing network. Peregrine Guard also provides auto-discovery of all devices making sure no device goes undetected and there are no unauthorized devices on the corporate network. Its central policy management and complete visibility and alert mechanism makes it unique in the market to alert and prevent spread of malwares which are now finding backdoor entry into corporate network via BYODs. Peregrine Guard also integrates well with Microsoft Exchange ActiveSync helping in remote wipe out of the device when required.

Central management of the P7 devices: Here we provide the master slave policy configuration facility to admin; from master peregrine 7 he can set the policy to different slave peregrine 7. put the all admin entered policy in XML and then convert that into ACL language (because we need to write policy into core switch).From master p7 send data(policy) to slave p7 through IP address and port number, and Master p7 will get the policies from all the p7 appliances through secure copy(SCP) and expect scripts. Master and slave will communicate with SCP.

II. PROJECT MODULES

1. *Device Fingerprinting*:

A device fingerprint is information collected about a device for the purpose of identification when device connected to organization through WIFI or LAN. Fingerprints can be used to fully or partially identify individual users or devices.

DHCP Fingerprinting: DHCP (Dynamic Host Configuration Protocol) client implementation of every operating system varies and these variations can be mapped to form a unique signature to identify a particular operating system. P7 uses following (but not limited to) DHCP signatures to identify an operating system uniquely.

HTTP Fingerprinting: HTTP (Hyper Text Transfer Protocol) Every HTTP communication sends an HTTP header called User Agent, which has information of the client. It is commonly like Operating system, device type, browser version, Brower SDK version. P7 maintains a database of well-known User agent formats and extracts relevant information to identify device attribute of the end device with HTTP protocol Fingerprinting, we get the device Attributes like OS Version of Device, browser information, apps etc..,

RADIUS protocol Fingerprinting: RADIUS (Remote Authentication Dial-In User Service) protocol encapsulates EAP messages. Extensible Authentication Protocol or EAP is an authentication framework frequently used in wireless networks and Point to point connections. EAP protocol carries information like Username, location and authentication tokes (MAC address of the wireless access point). By decoding RADIUS Protocols packets we get associate information like

Username/Group name, location with a particular device.

2. ***Centralized policy Configuration:***

This module helps admin to push common policies to the P7 box as well as individual policy. The centralized policy manager has the list of P7 appliances and their IP addresses, its login information and location of the policy.

Here we do two types of policy configurations:

1. Push policy
2. Pull policy

Pull Policy: Here we will get the policies from all the p7 appliances through secure copy (SCP) and expect scripts.

Push Policy: Here we will push the policy from master p7 to all the p7 appliances through SCP. This module will do remote copy policies in XML format from the centralized policy manager.

3. ***Centralized database management module:***

In this module master will have the details of all p7 appliances (IP, username, password and UUID). Master will get data from all appliances using Postgresql TCP (port 5432) and aggregate the data. Data will be managed using UUID of the appliance so that we can see the appliance based report as well. In centralized database will keep only the top 100 communications from each appliance. If user wants to see the detailed report he can connect to individual appliances through master.

4. ***Authentication module:***

There are two kinds of authentication. One is local authentication and another active directory authentication.

5. ***Local authentication:***

P7 Administrators are provided with facility of User Management. Administrators can add/remove users and their rights. If Local Authentication is used, user’s information is stored in the local database. Passwords are hashed using md5/sha2 algorithm and stored in the database.

6. ***Active Directory Authentication:***

P7 Administrators can choose to use Active Directory for user management. P7 Administrator has to provide Active Directory IP address and admin credentials to enable Active Directory authentication. Once Active Directory authentication is enabled every time a user tries to login, P7 will sends the credentials to Active Directory. If Active Directory validates and permits the user credentials, then the user is allowed to login to P7 User Interface.

7. ***Centralized reporting module:***

In this module we Collect data from all p7 boxes (slave) and we aggregate it, here we have different type of reporting. It is helpful to maintain hard copy of data about each device and what they access.

1. Security Reporting

Reports based on Intrusion Detection System alerts.

2. URL Category Reports

Peregrine Guard (PG) is a transparent appliance that sits behind wireless access points in enterprise networks. It passively analyses all the traffic coming from and going to all the mobile devices. This analysis leads to discovery of the type of device (iPhone / iPad /Android / Blackberry), OS version running in the device, user of the device, time of usage, location of device usage, some of the applications running in the device. Discovered information is presented in dashboards. IT will be able to define fine grained access control policies using the discovered attributes. Peregrine Guard applies these access control policies on the traffic between wireless devices and rest of the network (both intranet and the internet).

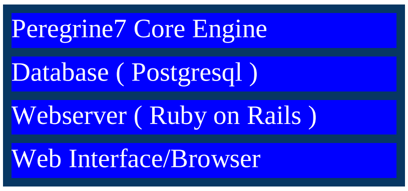

Fig1. Peregrine7 High-Level Product Architecture

Peregrine7 core engine:

Sniffs Wireless Traffic in the Enterprise and apply analytics to provide useful information about the devices (Operating System, Version, Type of Device, Authenticating User etc…). It also analyzes the traffic to derive security metrics like whether the device is Jail broken/Rooted, Device OS/ Application vulnerabilities etc..

Provides Internet traffic analytics, like what type of URLs the device is visiting. GeoIP analytics, Blacklisted IP database lookup etc… All the analytical data is stored into Database (Postgresql).

Database (postgresql):

All the data collected collected/analyzed by the core engine is stored into Postgresql Database.

Webserver:

Peregrine7 implements a Web User Interface in order to provide visibility into the collected data.

Web server runs on individual appliance and is implemented using “Ruby on Rails” framework. The Rails framework is connected to Postgresql database as it primary data source.
Web client:
Peregrine7 appliance administrator can login to the Web Interface using any browser. Peregrine7 provide rich user interface and uses latest UI tools like query, bootstrap etc...

III. SOFTWARE REQUIREMENTS AND HARDWARE REQUIREMENT

Software Requirements:
Operating system : Ubuntu 12.04 and above.
Software Tool : Ruby on Rails Framework, Java Script.
Database tool : Postgres.
Hardware Requirements:
RAM : 1GB
HDD : 20 GB
Processor : Pentium 4

IV. EXISTING SYSTEM AND PROPOSED SYSTEM

Existing System:

As mentioned in the Peregrine7 product architecture diagram. Peregrine7 provides control over the core engine as well as visibility into data collected by core engine using web interface. Every Peregrine7 appliance runs the Web Interface which can be accessed by Peregrine7 Administrator to configure the system or view reports/dashboards.

Proposed System (Peregrine7 Central Management System):

As discussed earlier, every Peregrine7 Appliance runs Web Interface that helps Administrators to configure and view reports/dashboards of Peregrine7 UI. However, for large Enterprises, there are multiple installations of Peregrine7 Appliances. It is cumbersome for the Administrator to manage multiple Peregrine7 appliances by individually logging into each of the appliance. Proposed Peregrine7 Central Management System solves this problem by providing a single interface to multiple Peregrine7 appliances/installations.

Challenges in the proposed solutions:
1. Management of multiple database connections.
2. Synchronization of configuration settings among multiple Peregrine7 boxes.
3. Data synchronization used for reports and dashboards.
4. High Availability.

V. MODULES (FUNCTIONAL AND NON-FUNCTIONAL)

1. Centralized database management module

In this module we have to manage multiple database connection from each P7 boxes.

2. Centralized reporting module

This module collects data from all the P7 boxes and aggregates it.

3. Centralized policy configuration module

This module helps admin to push common policies to all the P7 boxes as well as individual policy.

- Pull Policy
- Push Policy
- Synchronize Policy

4. Authentication module

There are two kinds of authentication.

- Local authentication
- Active directory authentication

5. Device Fingerprinting

Detect device attributes like Device Type (iPhone/iPad …), operating system (Android, iOS...), Operating System version, Device Model (HTC One, Sony etc...) by analyzing various network protocols

- DHCP fingerprinting
- HTTP fingerprinting
- sTCP/IP fingerprinting

VI. CONCLUSION

Peregrine Guard, its latest security offering for enterprises. It monitors BYODs (Bring Your Own Devices) in enterprise LANs and offers a complete security solution through few simple steps executed in seamlessly across wireless networks, without intruding on user privacy.

REFERENCES

[1]. [1] “Programming with World Wide Web” - by Robert W. Sebesta.
[2]. [2] wikibooks.org/wiki/Ruby_on_Rails
[3]. http://guides.rubyonrails.org/getting_started.html
[4]. http://www.fingerbank.org/
[5]. http://ruby.learncodethehardway.org/
[6]. http://railscasts.com/
[7]. http://www.net-square.com/httprint_paper.html
[8]. http://www.tutorialspoint.com/RubyonRails
[9]. http://www.w3schools.com/

[10]. http://getbootstrap.com/2.3.2/getting-started.html/

[11]. http://www.tutorialspoint.com/postgresql/