Mobile Vulnerability Assessment

Sujeet Metry[1] Avinash Avhad[2] Akshay Rathod[3] Prof. Rahul Patil[4]
[1,2,3]BE Student [4]Assistant Professor
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]BVCOE, Navi Mumbai

Abstract— The U.S. Department of Homeland Security, Office of Domestic Preparedness identified the need to examine and classify various types of vulnerability assessment methodologies, software, and tools as they would pertain to different types of assets. This Phase I study focused on the methodologies used to determine vulnerabilities and risks, which in turn, identify countermeasures that could be effective at reducing the risk by reducing the vulnerability. In identifying physical asset vulnerability assessment providers using a proven methodology, this study's goals were to develop criteria for analysis of various methodologies, clearly map capabilities and identify any capability overlaps, describe advantages and disadvantages of using particular methodologies, automated tools, software and technologies to assess different types of assets, and to provide evidence that methodologies, automated tools, software and emerging technologies can perform as advertised. Forty-four private methodologies were considered in this study. Sufficient information was found to make some level of assessment for 24 public (Federal, State, and local government) methodologies. Study findings drawn from this analysis include: (1) the most robust methodologies do not solely focus on one sector of the economy; (2) the quality of the assessor in all cases is very significant; (3) while all methodologies determined some calculate of risk, not many methodologies actually calculated a numerical value for that measure of risk; and (4) the training required to accurately use one of these methodologies diverse significantly in time and cost.

Key words: HTTP, URL, Cookie

I. INTRODUCTION

A flaw or weakness in system security procedures, design, implementation, or internal controls that may result in a security breach or a violation of the system's security policy. Vulnerability Assessment is the process of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities in a system.

Website security is today's most overlooked aspect of securing the enterprise and should be a priority in any organization. Increasingly, hackers are concentrating their efforts on web-based applications – shopping carts, forms, login pages, dynamic content, etc. Accessible 24/7 from anywhere in the world, insecure web applications provide easy access to backend corporate databases and also allow hackers to perform illegal activities using the attacked sites. A victim's website can be used to launch criminal activities such as hosting phishing sites or to transfer illicit content, while abusing the website's bandwidth and making its owner liable for these unlawful acts.

Vulnerability assessment is a crucial step in risk assessment, translating Hazard levels into Risk levels. Translating Hazard levels into Risk levels.

Hackers already have a wide repertoire of attacks that they regularly launch against organizations including SQL Injection, Cross Site Scripting, Directory Traversal Attacks, Parameter Manipulation (e.g., URL, Cookie, HTTP headers, web forms), Authentication Attacks, Directory Enumeration and other exploits. Moreover, the hacker community is very close-knit; newly discovered web application intrusions, known as Zero Day exploits, are posted on a number of forums and websites known only to members of that exclusive group. Postings are updated daily and are used to propagate and facilitate further hacking.

Web applications – shopping carts, forms, login pages, dynamic content, and other bespoke applications – are designed to allow your website visitors to retrieve and submit dynamic content including varying levels of personal and sensitive data. If these web applications are not secure, then your entire database of sensitive information is at serious risk. A Gartner Group study reveals that 75% of cyber-attacks are done at the web application level.

II. LITERATURE SURVEY

"25% of Vulnerabilities Found Involve Cross-Site Scripting"

The chart below shows the relative share of vulnerability types found and the sum equals 100%. Out of all vulnerabilities discovered XSS and Information Leakage are the largest share because they occur often and in some cases multiple times per application. Information leakage has the highest percentage increase which is up 7% from the 2012 level of 16% of detected vulnerabilities.

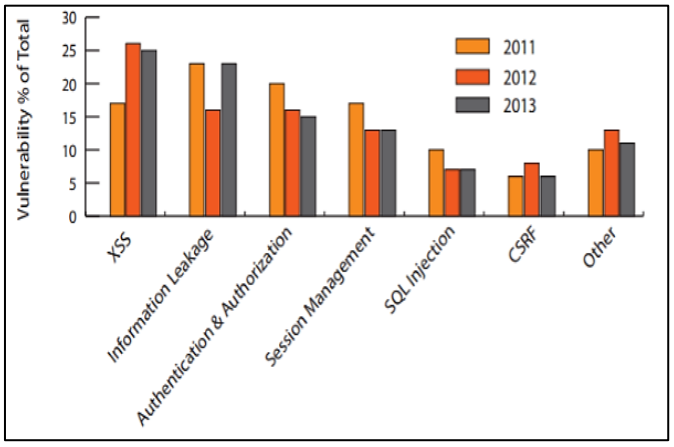

Fig. 1: 2013 Vs. 2012 & 2011 Web Application Vulnerabilities Found Trends

Figure 1: shows that three out of seven categories of vulnerabili¬ties declined in 2013 compared to 2012, while one category in¬creased and the remainder were essentially unchanged. Using the categories described above it was possible to identify areas within web applications that caused the highest number of vul-nerabilities during the period of study. In all three years, session management, cross-site scripting, information leakage, authenti¬cation &

authorization accounted for the highest number of vul¬nerabilities identified. None of the categories saw consistent increases each year. In¬formation leakage had the biggest percentage increase which nearly doubled in 2013 compared to 2012. This is likely due to accidental leakage of sensitive information through data trans¬mission or error messages. Authentication & authorization saw a third year in a row of de¬cline, albeit modestly. Whether this is because Authentication & Authorization are actually shrinking, or simply because other vul¬nerabilities are becoming more common is less clear.

General awareness within IT organizations about the importance of application security is growing. We see more of them demand¬ing that their vendors also take application security seriously. In the past, it was common to see vendors postpone security fixes in favor of releasing new versions of their software with new func¬tionality and unpatched vulnerabilities. The trend of consistent scanning, monitoring and correcting vul¬nerabilities is a process that many companies have implemented to reduce online risk. Information security teams who use reme¬diation data that integrates into Security Information and Event Management (SIEM) devices or Web Application Firewalls (WAF)can quickly patch vulnerabilities. While these trends are modestly good news, it is important to re¬member that vulnerabilities still exist across all categories. They exist in legacy applications and new applications. The threats from these vulnerabilities continue to evolve as cyber criminals experiment with new and different attack strategies.

A. Mobile App Vulnerabilities:

As more data is made available to mobile devices, mobile ap¬plication security grows in importance. The Cenzic Managed Services team has discovered the following vulnerabilities during 2013

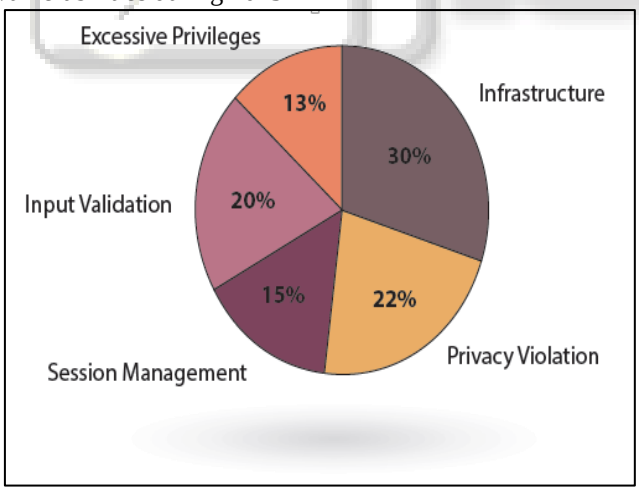

Fig. 2: Mobile Application Vulnerability Population

It is important to note that the largest category of risk (30%) comes from server configuration and patch level (Infrastructure), rather than the mobile application code itself. Web services must be secured.

Mobile developers should consider focusing their attention on how data is transferred to and stored on mobile devices as Input Validation (20%), Session Management (15%) and Privacy Violation (22%) combine to account for (57%) of mobile vulnerabilities. Storing unencrypted data on mobile devices is a significant cause for concern.

In the chart below we see that five vulnerability categories are extremely common in mobile applications. Privacy is not always seen as a security issue for the enterpise, though it may be for the user. Excessive privileges stems from developers giving their apps more power than is necessary to complete its function. Session termination is far too often an after thought

III. Prevention, Detection and Remediation

Proactive and consistent application risk management generally involves efficiently preventing, detecting and remediating vulner¬abilities across the entire application ecosystem. In general, early detection is more efficient but with some notable exceptions. In theory, prevention would be the most efficient way to reduce vul-nerabilities...except that no human developer can keep track of the 6,000+ known vulnerabilities. Nonetheless, coding best prac¬tices certainly help reduce vulnerabilities early on.

Similarly, Static Application Security Testing (SAST) tools can identify some vulnerabilities as the code is being written during development, but they create a large amount of superfluous infor¬mation and frustratingly high false positives which reduces their effectiveness. Some vulnerabilities simply cannot be seen until they are in a run-time environment. Dynamic Application Security Testing (DAST) tools tend to be much more accurate. As a result, many developers prefer to compile their code and dynamically test it in a run-time environment. In any event, since some vulner¬abilities only appear in a run-time environment, there must be a DAST scan prior to an application launch. We recommend sched¬uling and automating regular scans after going into production.

Remediation can follow several paths. While SAST solutions tend to point out each instance of a vulnerability that can cre¬ate unnecessary work, given that twenty instances of the same vulnerability can often be fixed with one high level code change identified by a DAST scan. Also sometimes SAST missed the 21st instance.

While it is often more efficient for developers to modify code while the project is still ongoing, that is not always the most efficient way to address a vulnerability. Web Application Firewalls (WAF), can create virtual patches to block vulnerabilities, sometimes in seconds. Given that new vulnerabilities are often discovered af¬ter the application goes into production, WAFs have a variety of advantages over the long haul.

Server Configuration incorporates aspects of prevention, detection and remediation. Some vulnerabilities disappear when the server is brought up to the proper patch level and configuration specifica¬tions.

A mature risk management model, whether it incorporates enter¬prise scanning software or managed services, provides both real time snap shots and trend data of your security posture over time. This allows both proactive and, if needed, crisis management ca¬pabilities. This is crucial not only to speed security response but to optimally reduce risk with inevitably limited resources.

IV. Proposed System

Android based intrusion detection system is an application of Cross domain platform over our Mobile Devices. Android

device is used to understand the common behaviour of user and used to update the user about the network status. Meanwhile if any of the node is identified to be the infected node from our intrusion detection system, Our Cloud based Vulnerability request to retrieve all the important data of user from mobile device via Android IDS application using Cloud.

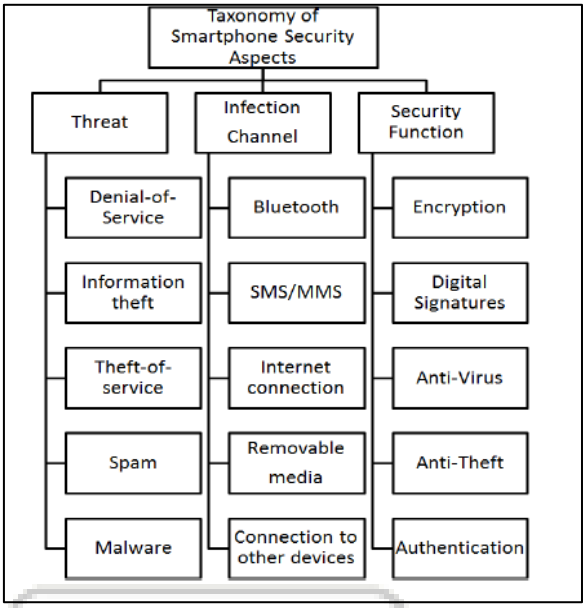

Fig. 3: Smartphone Security Aspects

Our cloud application send request to the android application after verifying the user id and create a temporary data space to store all the information into the cloud. All the information can be easily managed and monitored via Separate Web interface which is given to IDS Cloud Manager.

Once our manager identified any of the Blacklist user or misbehave with the mobile device, we directly block the specified user and restrict their file access

Further are the taxonomy for Smartphone security model. This model described about the measurementswhich are been used for providing security in Smartphone. It also includes the hierarchy in which Smartphone security goes with different applications of Smartphone. This model also suggests the security threats which can be considered for Smartphone in real terms. Model suggests us about the possible solutions available for security and possible threats to be prevented.

A. System Architecture:

1) Mobile Side Coding:

In mobile side coding a software is developed that has to be install in the android mobile of the user whom we have to track in real time Application The job¬¬¬ of software is continuously communicate with built-in Server.

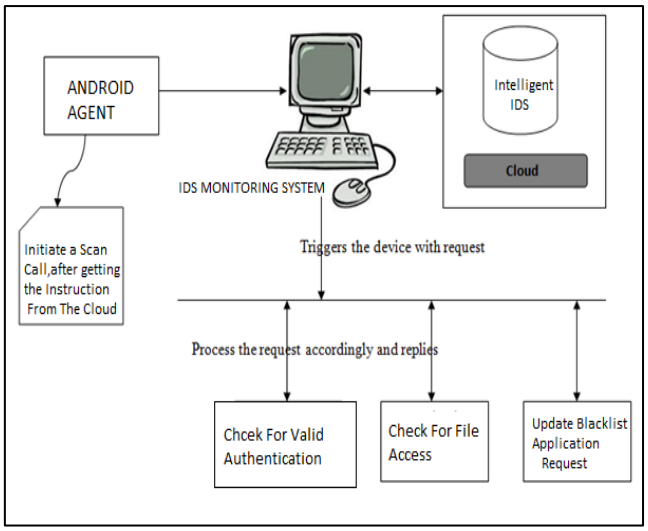

Fig. 4: System Architecture

2) PC Side Coding:

In PC side coding, there is an admin PC to monitor the system. The admin PC will receive the Status packet from the mobile device being traced. Whenever administrators want to track the person he just fined the mobile status in the database of the remote mobile. That will directly accepted by the admin PC and stored in the database along with time and date for future reference software automatically display current Status of the mobile application. Pc interface can directly check the user update and take any step for unauthorized file access, Authentication failure and Blacklist user from data access.

This publication is designed to assist organizations in implementing security patch and vulnerability remediation programs. It focuses on how to create an organizational process and test the effectiveness of the process. It also seeks to inform the reader about the technical solutions that are available for vulnerability re media.

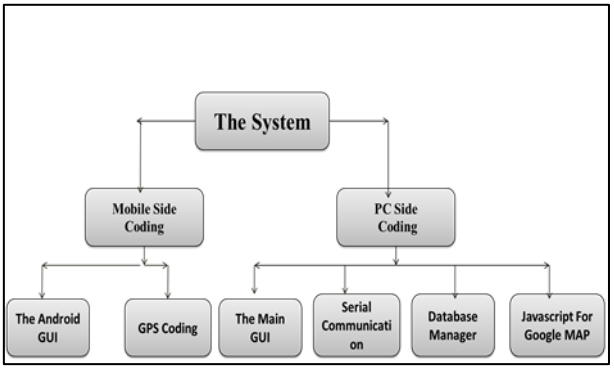

Fig. 5: Proposed System Block Diagram

V. CONCLUSION

We've identified the issues…now we have to fix them. 96% of applications have vulnerabilities with a median of 14 per application. Vulnerabilities per application is steady over the last three years. XSS, Leakage, Authentication and Session Management are most common. Privacy Violation and Excessive Privileges appear in over 80% of mobile apps. Enterprise software and managed services can make most efficient use of existing headcount. While the majority of corporations have the important security building blocks, such as firewalls and intrusion protection systems needed

for their security infrastructure, not enough organizations have comprehensive tools and practices in place for securing applications. The result is that hackers are increasingly focusing on and are succeeding with layer 7 attacks.

Application developers tend to focus on adding features rather than rooting out all application vul¬nerabilities. This combined with the daunting task of preventing, detecting and eliminating applica¬tion vulnerabilities explains part of the continued widespread discovery. Prevalence data should be coupled with objective risk scores, such as Conic's HARMTM Score, to optimally prioritize and reduce risk. A shortage of skilled application security professionals remains a major issue across the global business landscape. This is likely to continue and lead to the growth in managed security services as companies look to bring in specialists to augment their overextended teams. Ultimately this will lead to better protection for web, cloud and mobile applications.

REFERENCES

[1] [AA] S. Andersen and V. Abele, "Data Execution Prevention.Changes to Functionality MicrosoftWindowsXPServicePack2, Part3: MemoryProtection Technologies.,"http://www.microsoft.com/technet/ prodtechnol/winxppro/maintain/ sp2mempr.mspx.
[2] [ABL04] L.von Ann, M. Blum, and J. Langford, "TellingHumans and Computers Apart Automatically,"Communications of the ACM, Feb. 2004.
[3] [AL] Alexa, http://www.alexa.com/.
[4] [ASA+05] K. Anagnostakisy, S. Sidiroglouz, P. Akritidis, K.Xinidis, E. Markatos, and A. Keromytis. "DetectingTargetedAttacks Using Shadow Honeypots," in Proc. USENIX SecuritySymposium, August 2005.
[5] [ASLR] PaX Address Space Layout Randomization (ASLR).http://pax.grsecurity.net/docs/aslr.txt.
[6] M. Polychronakis, P. Mavrommatis, and N. Provos, "GhostTurns Zombie: Exploring the Life Cycle of Web-based Malware,"in USENIX Workshop on Large-Scale Exploits and EmergentThreats, 2008.

