International Journal of Combined Research & Development (IJCRD)
eISSN: 2321-225X;pISSN:2321-2241 Volume: 4; Issue: 10; October -2015

Cyber Security Challenges and Future

Nayana Hegde [1], Savitha N.G [2]

[1,2] Dept. of Electronics and Communication Engineering
Sri Krishna Institute Of Technology Bangalore, 560096

Abstract—the wireless communications infrastructure has been evolving rapidly in the last decades due to the need for more sophisticated information and data transfer. Mobile devices such as smartphones and tablets have become very popular in recent years. A mobile device may carry sensitive data and it is used in many personal and business transactions. Thus it has becomes an easy target for cyber criminals. Information security is main concern for every communication network. Throughput, exibility, and security form hackers is the main requirements for design of crypto engines. This paper describes the popular algorithms used in data security and their implementation using java crypto library.

Keywords- Network security, data transmission. Encryption, Decryption, Advanced Encryption Standard (AES), java Implementation

I. INTRODUCTION

Information security has become a major issue with the advance of wireless communication technology. There are several methods to provide security to the information that is being communicated. Four major security requirements are integrity, confidentiality, authentication and freshness [1]. Encryption is used to ensure confidentiality and message authentication code (MAC), functioning as a secure checksum, provides the data integrity and authentication in the network [2]. The IEEE 802.15.4 standard is intended to operate in unlicensed and international frequency band.802.15.4 specifies cryptographic procedures for protecting communications at medium access control layer. The MAC layer of 802.15.4

uses the AES-CCM protocol as security mechanism, it uses the AES algorithm as the core, uses CTR mode to ensure confidentiality of data, uses CBC-MAC mode to authenticate the public information of the header in MAC frame [3]. The data encryption algorithms used are generally divided into three major categories: Symmetric-key algorithms, asymmetric key algorithms, and hash algorithms. Asymmetric algorithms are more energy consuming. Hash functions, on the other hand, are typically used for verifying the integrity of the exchanged messages and may increase the transmission cost [4].

II. CHALLENGES

Four big challenges of cyber Security

- Security is hard work Rather than building a strong, Comprehensive security strategy, organizations are often enticed by shiny new tools that promise to be a magic

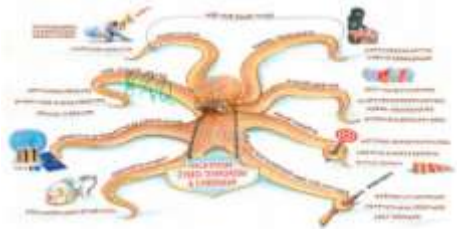

Fig. 1. Cyber Security

bullet to their security challenges. Instead, these organizations end up with a hodge podge of technologies from dozens of different vendors that don't integrate or provide visibility into the right data, leaving them vulnerable to hackers. It's time to focus on the basics. Building a solid foundation for good security is hard work. It starts with cleaning house, patching what needs to be patched, monitoring privileged users, identifying risks and reaching out to the right stakeholders. And as cloud and mobile technologies dominate the environment, expanding the number of endpoints and entryways into our systems and data, starting out with solid security is critical and no doubt harder than installing a shiny new tool.

- Attackers are extremely advanced Hackers have been quietly assembling themselves into complex criminal rings and hierarchies on the Dark Web, running extremely large, efficient and profitable operations that rival legitimate operations. More than 80 percent of cyber-attacks today come from these organized crime rings, according to the United Nations Office of Drugs and Crime.

This means that what we have typically considered protection firewalls, antivirus software and logging is simply not enough. In fact, breaches can lurk undetected in systems, even with these protective tools, for weeks or months, stealing information that causes maximum damage. To stop advanced threats, organizations must be able to: Perform advanced network and big data analytics to prevent unknown attacks, including those that use advanced malware. Detect both minor and major anomalies, such as traffic spikes on off hours or repeated

login attempts, across a wide range of data and network traffic. Respond extremely quickly once an initial security

Incident has been detected.

- We need the good guys to collaborate Hackers have advanced quicker than the good buys because they have tremendous collaboration with each other, sharing tools, expertise and knowledge around the globe rapidly. Unfortunately, the good guys are severely lacking in their own collaboration efforts. There's a lot of talk about sharing, but not a lot of action. According to analyst firm Enterprise Security Group, 65 percent of in-house security teams rely on numerous, and often unverified and untrusted sources for threat data. They have no way to recognize the incidents that actually need their attention, and they have no real-time visibility into emerging attacks.

  The private sector is hungry to collaborate on a global level to fight back, but we desperately need to break down our silos and cut through our current state of disorganization. We have seen this at my company, where we just made available to the public the 70 terabytes cyber threat data collected over more than two decades. In just the first 30 days of availability through the X-Force Exchange threat intelligence network, we've seen thousands of people logging in to understand, consume and share this intelligence and posting hundreds of new threat data contributions.
- There's a tremendous cyber security skills gap the good guys are outnumbered. Thirty-five percent of organizations have open security positions that they are unable to fill and 53 percent say it can take as long as six months to fill one need, according to The State of Cyber security: Implications for 2015, a study by ISACA.

  Until we recruit, train and retain enough highly skilled professionals, this gap will worsen. While industries are working harder to do this, and large tech companies are partnering with universities and talent pools on skills training, 81 percent of security leaders still say staffing challenges will stay the same or get worse in the next five to 10 years.

III. Tools and Techniques

There are a lot of open source tools available to counter these threats so that your device is not at risk.

1) OSQuery
   osquery allows you to easily ask questions about your Linux and OSX infrastructure. Whether your goal is intrusion detection, infrastructure reliability, or compliance, osquery gives you the ability to empower and inform a broad set of organizations within your company.
   source:http://osquery.io
2) Security Onion
   Security Onion is a Linux distro for intrusion detection, network security monitoring, and log management. It's based on Ubuntu and contains Snort, Suricata, Bro, OSSEC, Sguil, Squert, Snorby, ELSA, Xplico, NetworkMiner, and many other security tools. The easy-to-use Setup wizard allows you to build an army of distributed sensors for your enterprise in minutes
   source:http://blog.securityonion.net/p/securityonion.html
3) Skyline
   Skyline is a real-time* anomaly detection* system*, built to enable passive monitoring of hundreds of thousands of metrics, without the need to configure a model/thresholds for each one, as you might do with Nagios. It is designed to be used wherever there are a large quantity of high-resolution timeseries which need constant monitoring. Once a metrics stream is set up (from StatsD or Graphite or other source), additional metrics are automatically added to Skyline for analysis. Skyline's easily extendible algorithms automatically detect what it means for each metric to be anomalous. After Skyline detects an anomalous metric, it surfaces the entire timeseries to the webapp, where the anomaly can be viewed and acted upon.
   source:https://github.com/etsy/skyline/
4) Google Rapid Response// GRR consists of an agent (client) that can be deployed to a target system, and server infrastructure that can manage and talk to the agent.// source:https://github.com/google/grr/
5) OSSEC
   OSSEC is an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response.
   It runs on most operating systems, including Linux, MacOS, Solaris, HP-UX, AIX and Windows.
   Check out OSSEC features and how it works for more information about how OSSEC can help you solve your host-based security problems.
   source:https://www.ossec.net
6) Scumblr and Sketchy Scumblr is a web application that allows performing periodic searches and storing / taking actions on the identified results. Scumblr uses the Workflowable gem to allow setting up flexible workflows for different types of results.
   source:https://github.com/netflix/scumblr/
7) RAPPOR
   RAPPOR is a novel privacy technology that allows inferring statistics about populations while preserving the privacy of individual users.
   This repository contains simulation and analysis code in

Python and R.
source:https://github.com/google/rappor/

8) OpenVAS
OpenVAS is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution.
source:http://openvas.org/

9) OpenSSH
OpenSSH is a FREE version of the SSH connectivity tools that technical users of the Internet rely on. Users of telnet, rlogin, and ftp may not realize that their password is transmitted across the Internet unencrypted, but it is.

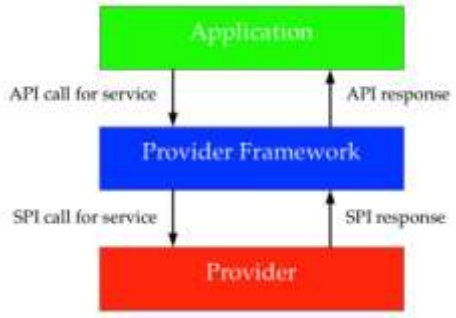

Fig. 2. JCA Architecture

OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, OpenSSH provides secure tunneling capabilities and several authentication methods, and supports all SSH protocol versions.
source:http://www.openssh.com/

10) MIDAS
MIDAS is a framework for developing a Mac Intrusion Detection Analysis System, based on work and collaborative discussions between the Etsy and Facebook security teams. This repository provides a modular framework and a number of helper utilities, as well as an example module for detecting modifications to common OS X persistence mechanisms.
source:https://github.com/etsy/MIDAS

IV. IMPLEMENTATION

A. Encryption and Security through Java Cryptography Architecture

For a long time, Java has provided security-related functions. Among the security-related functions, Java Cryptography Architecture (JCA) is the core one. JCA uses a provider structure with a variety of APIs related to security. These functions are essential for modern IT communication encryption technology, including Digital Signature, Message Digest (hashs), Certificate, Certificate Validation, creation and management of Key, and creation of Secure Random Number. Few Available JCA Providers:

- SUN 1.7
- SUNEC 1.7
- SUNJCE 1.7
- SUNSASL 1.7

Architecture Of JCA Adding JCA Service Providers

JCA security providers can be added:
As Part of Java SE for all application to use
As part of a single application

Adding a provider in a JAVA Se installation requires:
Writing the provider's JAR file to:
¡java-home¿/lib /ext [5] Encryption and decryption are fundamental requirements of every secure-aware application, there-

Fig. 3. Encryption of XTEA

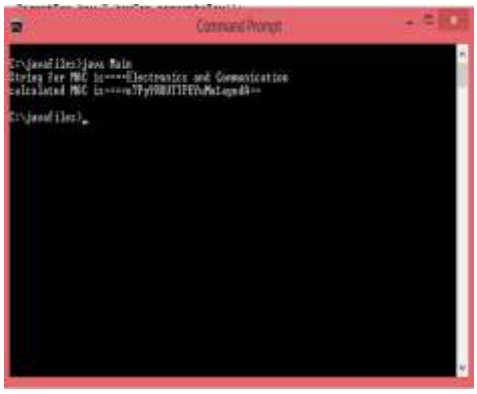

Fig. 4. Output of MAC

fore the Java platform provides strong support for encryp- tion and decryption through its Java Cryptographic Exten- sion (JCE) framework which implements the standard crypto- graphic algorithms such as AES, DES, DESede and RSA. This tutorial shows you how to basically encrypt and decrypt files using the Advanced Encryption Standard (AES) algorithm. AES is a symmetric-key algorithm that uses the same key for both encryption and decryption of data.

Basic Steps Here are the general steps to encrypt/decrypt a file in Java:

1 Create a Key from a given byte array for a given algorithm.

2 Get an instance of Cipher class for a given algorithm transformation.

3 See document of the Cipher class for more information regarding supported algorithms and transformations.

4 Initialize the Cipher with an appropriate mode (encrypt or decrypt) and the given Key.

5 Invoke do Final (input bytes) method of the Cipher class to perform encryption or decryption on the input bytes, which returns an encrypted or decrypted byte array.

6 Read an input file to a byte array and write the en- crypted/decrypted byte array to an output file accordingly.

B. Results

Implementation of security scheme is done using java pro- gram.Figure 3. shows the encrypted results of Tiny encryption algorithm. Figure 4. shows the result of MAC generartion.Also figure 5. shows the message digest generation.

Fig. 5. Output of MAC

CONCLUSION

Security has become a major concern in the wake of all recent high profile break-ins and the resulting outcry. Protecting web applications is not only security imperative, it is also good business. It is now critical to maintain a safe and reliable world, and we have to implement security.in our systems, networks and devices. Encryption is alone not enough to ensure the confidentiality, authenticity and integrity of communicated data. AES encryption algorithm is adopted by many security standards such as IEEE 802.15.4 and IEEE 802.15.6. AES algorithm which provides data en- cryption also provides additional security services such as data authentication, integrity when it is combined with appropriate functionality.

REFERENCES

[1] H. M. H. Xueying Zhang and C. Li, "Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks," *Biennial Symposium on Communications IEEE*, pp. 168–172, 2010.
[2] K. V. n. Ch.P.Antonopoulos, Ch.Petropoulos, "The effect of symmetric block ciphers on wsn performance and behavior," *International Workshop on Selected topics in Mobile and Wireless Computing*, vol. 2, no. 4, pp. 799–807, 2012.
[3] H. H. B. Feng, "Parallel and multiplex architecture of aes-ccm coprocessor implementation for ieee 802.15.4," Sept. 2013.
[4] H. Y. Soufiene Ben Othman, Abdelbasset Trad, "Performance evaluation of encryption algorithm for wsns," *IJNSA*, vol. 2, no. 3, pp. 52–63, July 2010.
[5] J. Lawson, "Java cryptography," *Cogent Logic Limited United Kingdom*, 2013.