Security Vulnerabilities in Open Source Projects: An India Perspective
Krishnashree Achuthan¹, Sreekutty SudhaRavi¹, Renuka Kumar¹, Raghu Raman ²

¹Amrita Center for Cybersecurity Systems & Networks
Amritapuri, Amrita Vishwa Vidyapeetham, Kollam, India
{krishna@amrita.edu, sreekuttysr555@gmail.com, renuka@am.amrita.edu}
²Center for Research in Advanced Technologies for Education
Amritapuri, Amrita Vishwa Vidyapeetham, Kollam, India
{raghu@amrita.edu}

Abstract — Educational and governmental organizations are heavy users of Free and Open Source Software (FOSS) due to the numerous economic advantages it offers. But because of the lack of formal notification of vulnerabilities in them these users are left with exploitable risks in their systems with known vulnerabilities which could completely offset the economic gains and lead to unrecoverable losses. India is one of the largest consumers of Free and Open Source Software (FOSS) though in the last few years there has been concerted effort to contribute to the movement as well as create its own FOSS to support local languages. This paper compares and analyses the public disclosure of vulnerabilities in Free and Open Source Software (FOSS) to those of non-open source systems. Our case study with (N=218) Information Technology (IT) professionals working in computer systems, networks and application development areas indicates an urgent need to enhance vulnerability handling practices for Free and Open Source Software based applications. This study has interesting implications for Information and Communications Technology (ICT) policy makers in the government as well as private sector who are increasingly advocating the use of FOSS.

Keywords — Vulnerability; Open Source Vulnerability Database; National Vulnerability Database; Cybersecurity threats; Free and Open Source Software

I. INTRODUCTION

With the dawn of the internet, the advances in computing and software development have grown exponentially in the past two decades. These advances have brought about solutions to complex problems in diverse areas from astrophysics to biological systems, automation, precision etc using a variety of computing systems and software. In spite of their need, these software systems are, inherently vulnerable. Vulnerabilities have been a major source of cyber threats in the modern era of computing. According to Cowan [1], because of the availability of the source code hackers do not have to spend time on black box testing for introducing security vulnerabilities but there is no clear research evidence that closed source development provides better application security. Fixing losses from vulnerabilities such as data leakage, espionage, compromise of systems and processes is a fairly costly affair. A recent study indicated that the presence of vulnerable environments and rise of sophisticated attacks can cost the world economy up to 3 trillion USD by 2020 [2]. To study the importance and impact of cyber threats, with pro-active measures to counter them, organization such as CERT (Computer Emergency Response Team) teams have been established in various countries. In addition, since a cyber-attack can come from multiple sources of compromise, cataloguing vulnerabilities has become a practice. These catalogues exist in the form of vulnerability databases, security bulletins, blogs etc. However, these databases cover vulnerabilities of a subset of common softwares.

Adoption of free and open-source softwares (FOSS) has been on the rise due to its free availability and flexibility in terms of its licensing, modification and usage. The wider peer review of open-source software and independence from relying on a single software entity (person, group or firm) is often considered secure and sustainable approaches to software development. Thus many governmental initiatives, educational and research institutions as well software vendors use open-source tools in development of multiple applications.

The Indian government has embraced and propagated the use of FOSS in all its initiatives. Two major initiatives that have been established are National Resource Center for Free and Open Source Software or NRCFOSS [3] by Central Government and International Centre for Free and Open Source Software, ICFOSS [4] Kerala State Government. The past five years have seen large scale government projects completely rely on FOSS frameworks. The Bharat operating system solutions or BOSS [5] was an initiative to provide indigenous GNU/Linux solution with multi-lingual support to the Indian community of developers and beneficiaries. Aadhar is one of the largest open source technology (OST) based project driven by the Government of India with the goal of providing unique identification to 1 billion people. Use of OST allows maintenance of vendor neutrality. Also achieving the scale is done using techniques such as data sharing to avoid total dependency on single vendor. This reduced dependencies on any single program, yet was able to achieve the massive requirements in data storage, data security etc.

One of the concerns with using open-source tools is that they lack security as opposed to closed source softwares that provide periodic security updates. In many cases due to modifications done to the source files, effort involved in
patching insecure code is often absent. Thus although there is much more awareness of the grave consequences from cyber-attacks in recent years, many organizations are often confused and/or take precautionary measures without a detailed analysis of their specific needs which may or may not use open-source tools. This can cause either a larger than necessary expenditure on anti-hack preventative solutions or investments in less probabilistic attack scenarios.

The drive to adopt free and open source software (FOSS) has been so successful in that more than fifty percent of world’s largest enterprises have adopted them. However this also implies presence of large amount of vulnerabilities due to incorporation of binaries that are difficult to patch. [aspect]. A few examples of FOSS libraries and frameworks that are downloaded the most include Struts 1.X, Google Web Toolkit, Hibernate etc. Beyond large enterprises academic institutions are also heavy users of FOSS [6,7,28]. On the other hand, there are also large projects that build systems from licensed software [8,9].

The rate of vulnerability disclosures have increased steadily every year. However there is a large variation in the statistics reported by vulnerability databases such as National Vulnerability Database[10] or NVD and Open source Vulnerability Database i.e. OSVDB [11]. Enumerating the criticality and rating of the most vulnerable systems are available from a variety of sources [12,13]. Since most of the data that goes into these databases are consolidated by approximately 760 vendors, a comprehensive view of the state of vulnerabilities and security threats and its relevance to majority of developers and organizations is absent. Even more distant is its perceived usefulness to FOSS developers with less than 25 % representation of FOSS systems in these vulnerability databases. Since the area of cyber security is transforming more into cyber physical system security, the vulnerabilities of physical systems connected to the internet is just beginning to emerge. The lack of standardized publication of FOSS vulnerabilities, and absence of the eco-system that non-FOSS alternatives have developed over the years i.e. automated alerting and updates is a huge lacuna in the cyber space. This has led to the increase in exploitation of existing vulnerabilities that require critical fixes. Even when vulnerabilities are known, there is little done towards establishing a structured process to handle them which could prevent further exploitation.

In this paper a literature survey of vulnerabilities with emphasis on FOSS vulnerabilities is presented. Following this an analysis of popular vulnerability databases and the representation of FOSS vulnerabilities in them is given. Details of National FOSS projects is elaborated. Paper concludes with recommendations on building of information database that has the capability to capture vulnerabilities and counter measures for FOSS and domestic developers that is currently absent.

II. RELATED WORKS

The area of computer science and its applications have plenarily transmuted the lifestyle of humans. Software application development and provision of services is one of the largest global commercial industries. The growth of open source initiative and establishment of vulnerability databases has brought a cohesive approach to software development by sharing of resources and information. Most definitely there are many merits to this approach such as creating a vast network of developers build the system on the one side and on the other side providing broader understudying of the architecture to enhance their individual skills.

However vulnerabilities continue to exist as part and parcel of software development processes. Ng et. al [14] point out the existence of latent vulnerabilities when code is re-used as is the case with many customized applications that call for third party libraries. An important consideration is that because hackers have immediate access to the source code of the FOSS, Levy [15] felt that it is easier to introduce zero day vulnerability in FOSS that then can be exploited later. Alhazmi [16] has proposed Vulnerability Discovery Models but it is not entirely clear if the model will yield different results for FOSS vs. closed source systems.

When vulnerabilities are discovered in these libraries, software update managers don’t always pick up modified symbolic links to these codes and thus resulting in un-patched code being retained and referenced. In case of open source softwares, such automated updates are also absent. Rahimi et. al [17] use the code characteristics in itself to predict its vulnerability. They show that measures for code complexity and code quality i.e. compliance to secure coding rules can provide alternate strategies to predicting its vulnerability without external databases. However the issue here is that this is only valid for those cases where source code is available i.e. open source software or self-built ones. Although this does not represent the licensed obfuscated spectrum of code, there is merit in considering its usage to gauge the extent of vulnerability. Not all organizations invest in dedicated teams pursuing discovery of vulnerabilities in open source stacks. They either look for them in databases such as OSVDB [18] or CVE (common vulnerability exposure) [19] etc. There are a number of publications that have looked at such vulnerability databases in detail specifically for 1) understanding the growth of vulnerabilities and 2) the impact they might have.

Open source libraries are abundant with vulnerabilities. The Schryen [20] study shows that the mean time between vulnerability disclosures was lower for open source software in half of the cases, while the other cases showed no differences. Lack of compliance to secure coding practices, and monitoring in addition to absence of incorporation of notified vulnerabilities in a standardized way leads to open vulnerabilities remaining for relatively long periods of time. Vulnerabilities and security issues can be eventually used
for illegal activities. Payne [21] highlights the importance of having the peer review process while Levy [15] argues that despite the source code availability no one is actually reading it. Recent reports [22] have revealed that applications that use 30 or more libraries with vulnerabilities compromise over 80% of their code to malicious attacks. The loss can vary from data leakage or corruption, to complete loss of control. Analyzing a survey from 3500 developers using open source libraries, one of the important findings [23] has been that 57% of the organizations using them lack any policy and 76% of the organizations lack useful controls over open source usage. In addition, another survey [24] found that 42% of large organizations do not provide any ongoing security awareness training. Interestingly, the survey indicates that 93% of organizations where security training was not adequately provided had staff-related breaches compared to only 47% in organizations where training was effectively done.

III. THE FOSS ERA OF INDIA

One of the largest consumers of FOSS, India is making rapid strides in adopting FOSS across domains like eGovernance, Education, Health and most recently into computing. The common FOSS components in these applications are – MySQL, JBOSS, J2EE, Tomcat, Struts, Hibernate, and Hadoop.

Table 1: FOSS Adoption in India

<table>
<thead>
<tr>
<th>Domain</th>
<th>Project</th>
<th>Project details</th>
</tr>
</thead>
<tbody>
<tr>
<td>eGovernance</td>
<td>Aadhar</td>
<td>Aadhar is a 12-digit individual identification number which will serve as a proof of identity and address, anywhere in India. With 60 million issued the project is handling up to 1 million applications per day.</td>
</tr>
<tr>
<td></td>
<td>e-Panchayat</td>
<td>e-Panchayat is an e-Governance initiative for rural sector providing comprehensive services for Gram Panchayat functions; Over 2.36,210 dynamic websites created.</td>
</tr>
<tr>
<td>Education</td>
<td>IT@Schools</td>
<td>World’s largest simultaneous deployment of FOSS based ICT education covering over 12,000 schools, 6 million students and 2 million teachers.</td>
</tr>
<tr>
<td>Health</td>
<td>OSDD</td>
<td>Open Source Drug Discovery (OSDD) is a CSIR led team India Consortium with global partnership with a vision to provide affordable healthcare to the developing world.</td>
</tr>
<tr>
<td>Compute</td>
<td>Parichit</td>
<td>Parichit is open source OCR for Indian languages. The project aims to create high quality training data for creating tesseract language models.</td>
</tr>
</tbody>
</table>

IV. RESEARCH STUDY

A. Comparison of Security Vulnerability Databases

There are several databases that catalog software vulnerabilities. Table 2 shows the most prominently referred to databases. The table compares their key characteristics. These databases vary in their emphasis of type of products that are reported upon, the verification processes used, the technical depth of information, the sources that are credited. The common thread across most vulnerabilities is the reference to a CVE number (common vulnerability exposure) that serves as a dictionary of all vulnerabilities. A comparison of these databases reveal OSVDB as the most comprehensive one amongst them with over 100,000 vulnerabilities listed in them. The database does covers cross-domain software and hard ware related vulnerabilities in web applications, mobile devices, scada systems etc. There are other databases such as the exploit databases that information related to exploits.

Table 2: Comparison of Vulnerability Databases

<table>
<thead>
<tr>
<th>Database</th>
<th>CVE</th>
<th>OSVDB</th>
<th>NVD</th>
<th>Exploit Database</th>
<th>Security Focus</th>
<th>Secunia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Data</td>
<td>Vulns</td>
<td>Vulns</td>
<td>Vulns</td>
<td>Exploits</td>
<td>Vulns</td>
<td>Vulns</td>
</tr>
<tr>
<td>Total Records</td>
<td>59508</td>
<td>101062</td>
<td>59548</td>
<td>27536</td>
<td>60030</td>
<td>49525</td>
</tr>
<tr>
<td>Data Field</td>
<td>6</td>
<td>15</td>
<td>14</td>
<td>10</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>Update frequency</td>
<td>Varies</td>
<td>Varies</td>
<td>Hours</td>
<td>Varies</td>
<td>Daily</td>
<td>Daily</td>
</tr>
<tr>
<td>CVSS score</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Software Vulns</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Web App Vulns</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SCADA Vulns</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Mobile Vulns</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Data source:

| Vendors, Mailing lists, Advisor is etc | Vendors, Mailing lists, Advisor is etc | CVE, Security checklists | Online submittals, mailing list | Vendors, Mailing lists, Advisor is etc | Vulnerability intelligence solution |

We conducted a survey with 324 people related to the IT services management at educational, medical and government institutions. The survey was administered online and 218 people responded to the survey.

V. RESULT ANALYSIS

A. Most Popular Open Source Software in India

A survey of most commonly downloaded softwares by India are tabulated in Fig. 1. This data was obtained from download statistics of sourceforge.net.
B. Analysis of Vulnerabilities in Open Source Software

The number of vulnerabilities in FOSS in the most common vulnerability databases such as CVE, OSVDB, NVD and Secunia are shown in Fig. 2. The yearly average of vulnerabilities ranged between 10 and 65 and there is a large variation amongst them.

On analyzing the source code of several of them using static code analysis, there were glaring issues such as buffer overflow, access violation, common errors and system version compatibility. Using source code analysis tools such as Findbugs [25], Flawfinder [26] etc. an initial assessment of vulnerabilities were done. This was followed by manual verification. Based on our evaluation, around 163 out of the 213 program files had vulnerabilities. The reported vulnerabilities were obtained from various databases such as CVE, OSVDB, and NVD etc. In many cases the number of real vulnerabilities are about 10 times higher than the reported (Fig. 3. only for Apache HTTP Server).

C. SCADA Vulnerability

Physical devices that are part of control systems have been found to be inundated with security issues [27]. This is primarily due to the use of FOSS that are old and with no mechanism of patching when deployed amongst civilian population. In addition, due to the fragmented nature of the source components, plugging the security vulnerabilities becomes nearly impossible. With the recent policy directive from the Indian government that peripheral devices like printers, routers etc. must work with FOSS operating systems, so there is an urgent need to educate the IT professionals about SCADA vulnerabilities. In light of this, it becomes important to publicize the presence of these vulnerabilities and catalog them in popular databases. Fig. 4 shows the listing of SCADA vulnerabilities.

It was astonishing to observe that practically no one was aware of the worldwide initiatives like OSVDB, NVD and CVE which has more than 200,000 vulnerabilities documented. (Fig. 5).
Equally interesting is the fact that close to 50% of the people surveyed are neutral about vulnerability of the open source vs. closed source tools. (Fig. 6.)

Given the huge uptake in FOSS usage in various domains like Health, Education and eGovernance in India, it is consistent that a huge majority of respondents would like to see an India wide vulnerability database. (Fig. 7.)

Based on the survey data over 43% of the respondents are neutral about upgrades to latest software. (Fig. 8.)

VI. CONCLUSION AND FUTUREWORK

Indian institutions i.e. the Government, academia and industry have recognized the importance of FOSS and promoted its usage in a large scale. India’s e-governance projects alone have over 30 mission mode projects that embrace FOSS affecting millions of citizens with respect to their health records, education, financial transactions etc. With proliferation of FOSS clubs, India is emerging to become one of the largest FOSS hubs. One of the most critical aspects of FOSS is in ownership of how its vulnerabilities should be handled. A standardized process and policy could go a long way to adoption of “secure FOSS” (SFOSS). The source hardening project in the US [28] is a directed effort towards improving the open source code by the Department of Homeland security. The economic benefits from a SFOSS are staggering and could completely change the dynamics of how businesses, governments and individuals operate. CERT-In [29] has made several public advisories in newspapers and other media avenues to educate vulnerabilities of existing popular softwares, which is very helpful and step in the right direction. A governance model for handling vulnerabilities in FOSS including advocacy by academia, govt. and private players is the need of the hour. A comprehensive information database with both software, hardware covering both FOSS and non-FOSS systems – a single repository - is the need of the hour. The government should play an active role in creating the right policies and a conducive environment to capture, analyze, publish and securely propagate the vulnerabilities and countermeasures. This has to be done with encrypted communications and authenticated users. A systematic approach to acknowledge usage, provide updates, authenticate vulnerabilities overseen by a neutral multi-party consortia including ISPs and open source distributors is required to be in place. Initiatives like NRCFOSS, ICFOS should run campaigns to sensitize our engineers and IT related professionals to the importance of proactively upgrading to the latest versions of the FOSS components in their application stack.

One of the most important features that this platform could provide is the same level of services to domestic developers and their customers that are not global.
audience. Timely resolution of zero-day vulnerabilities, minimizing critical exploitation due to lack of awareness and due-diligence would be important metrics to consider.

ACKNOWLEDGMENT

Our work derives direction and ideas from the Chancellor of Amrita University, Sri Mata Amritanandamayi Devi. The authors would like to acknowledge the contributions of faculty and staff at Amrita University whose feedback and guidance was invaluable.

REFERENCES