Advanced Risk Mitigation of Software Vulnerabilities at Research Computing Centers

Urpo Kaila
CSC - IT Center for Science Ltd.
Finland
urpo.kaila@csc.fi

Abstract— **Software security vulnerabilities have caused research computing centers concern, excess work, service breaks, and data leakages since the time of the great Morris Internet worm in 1988. Despite evolving awareness, testing and patching procedures, vulnerabilities and vulnerability patching still cause too much trouble both for the users and for the sites.**

In this paper we will analyze operational risks based on software vulnerabilities, evaluate operational costs for deploying released security patches, and identify the benefits of deploying the updates. As sources for our study we use vulnerability and incident metrics, interviews, and an international survey targeted to system administrators and security experts at research computing centers.

The paper recommends ways on how sites and providers could improve technology, procedures and best security practices, such as tuning rolling reboots, improving site specific risk identification, and implementing more fine grained access controls, and intrusion detection measures.

Keywords - Software Vulnerability; Security, Risks, Availability

I. INTRODUCTION

In January 2017 a total of 81 424 vulnerabilities had been assigned a Common Weakness and Exposure Identifier (CVE ID) since 1999. Some interesting statistics is also available about the distribution of different vulnerability types [1]. According to statistics on Linux kernel vulnerabilities, also since 1999 [2], 16.2 percent of these were related to gathering unauthorized information, and 14.0 percent to gaining unauthorized access.

Despite the fact that hardware providers and software developers have already greatly improved their capabilities of mitigating vulnerabilities [3], software vulnerabilities still pose major risk in particularly for research computing centers, which provide services based on many different software sources on different maturity levels. Also, resources and procedures for vulnerability management at sites, might not be optimal.

In a previous study [4] we have shown, that security vulnerabilities in software and related patching, can result in considerable downtime for users. If an unpatched vulnerability is exploited, severe impacts on confidentiality, integrity, and availability could follow. Possible outcomes can include loss of system integrity, exploiting user credential, data leakage, prolonged downtime for services, loss of stakeholder trust for sites and providers, and legal actions including compensations for damages.

In addition to well known standard risks, such as hostile networks scans for known vulnerabilities or brute-force password guessing, sites must also mitigate hard to detect risk of advance persistent threats (APT) related to surveillance, cyber warfare or activities by advanced criminals. These kind of risk can use a vast array of methods from stealth technologies, social engineering and attacks trough internal trusted systems and networks.

Fortunately, severe systems compromises seem to be rare, at least among research computing centers, but risk identification and mitigation could still be greatly improved at reasonable cost in terms of resourcing and loss of usability, as we will show in this paper.

Our objective is to present a fact based review of realized vulnerability related incidents from the viewpoint of research computing centers. We will compare handling and outcome of different type of vulnerabilities at different type of sites. Based on our findings, we will suggest practical improvements for advance risk mitigation to improve system security without excess loss of usability or need for excess additional resourcing.

For a fact base review, we will use several sources of information. We start from public information of CVE's and Vendor Bulletins.

Our methodology is based on well known best practices for information security, as defined in international standards, such as ISO/IEC 27001 and in advisories such as NIST publications on computer security and cybersecurity practice [5].

A big challenge in studying risk and security management is, that a lot of crucial information is confidential. In contrast, much skewed information or even disinformation is also available in abundance, and it is hard to distinguish disinformation on security from facts. Vendors, cyber authorities, vulnerability researches have all their own overt and covert motivations to disseminate information. Additional, information security is a topic which is often experienced emotionally in terms of fear, confusion, and exaggerated urgency.

In our study, we will approach peers and peer sites in form of a trusted surveys and interviews. We will identify and exclude such truly confidential information which peers

cannot disclose even under non-disclosure agreements, but aim to gather other aggregated substantial information under transparent anonymity.

In addition to a survey and interviews, we will also analyze technology. Current and state-of-the art related network and systems security controls will be discussed shortly related to current implementations at sites, based on public information. Different computer security contexts will be discussed separately: User identification, authentication and authorization; protecting the login nodes for computing services; access controls for of user space; protecting access and integrity of infrastructure and system administration; intrusion detection and incident management.

We aim present our results in a such form, that our peers and stakeholders could directly use them as facts to facilitate their own decision making in vulnerability and risk management.

Finally, we will discuss possible next steps on how the communities and the providers could proceed to improve vulnerability risk mitigation internally and in joint actions.

II. MATERIAL AND METHODS

In this chapter we define the terms, material, and methods used in this paper.

The objective of this paper is to identify best practices on how advanced risks related to to software vulnerabilities can be mitigated at research related computing centers. This is a personal mission of the writer, but also a common concern for most research based computing centers.

According to CVE, a vulnerability is a weakness in the computational logic (e.g., code) found in software and some hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Also other types of vulnerabilities exist, which might or might not depend on software vulnerabilities: hardware vulnerabilities, insecure configurations, inadequate operational security, flaws in security architecture, or deficient security awareness. Lack of management involvement and support for security also creates indirectly a breeding ground for security vulnerabilities.

Risk has traditionally been defined as probability of loss or harm, but currently also probability of positive outcomes is included in definition of risks. In risk management, risks are basically to be identified, assessed and treated. Various sources of information are to be used for risk identification, depending on context, also many frameworks are available for risk assessment. A probability based approach is perhaps the most common, but for well defined environments for example fault-tree analysis can also be used for risk management.

Risks can by textbook examples be treated by risk avoidance, risk outsourcing, risk retention, and/or risk mitigation. Avoiding risks can be done avoiding a vulnerable activity, laptop thefts from cars can for example be avoided by banning staff to take laptops out of office – a decision with can can have negative consequences for productivity, usability, and reachability. Outsourcing risks can be done for example by insuring property and risk retention means simply taking the risk by, for example, budget for some wastage. In this paper we focus on risk mitigation, on how risks can reduced in an optimal way in terms of security, usability and efficiency.

Here we focus primarily on information security risks, which can be mitigated by security controls. The security controls include technical controls, such as patching vulnerable software, or configuration and change management in a wider sense, but it is good to keep in mind that there exist also other categories of security controls. A typical categorization of information security controls lists technical/managerial, reactive/proactive, detective/preventive/reactive, and compliance controls.

A very clear but perhaps an oversimplified presentation of the relations between threats, weaknesses (vulnerabilities), security controls and impacts has been presented by the OWASP Foundation [6], which supports software (although primarily web based software) safety and security.

Finally, our last term in the paper objective, research based computing centers, refers in this paper to national computing centers for academic research and computing centers for research infrastructures. These computing centers typically provide shared resources for supercomputing, high performance computing, and/or grid computing. The IT environment is mostly based on Linux or other unix based systems and the user interface, from which the users can submit jobs, is typically shell based although graphical user interfaces also exists. An updated list of the most efficient supercomputers for massively parallel computing can be found from the "Top 500 list "[7].

As material and methods for this paper we will study a couple of vulnerability cases, perform a survey to research computing centers and interview system administrators.

From the vast amount of published vulnerabilities, only a few can be selected as examples. Most published vulnerabilities don't even effect substantially or at all computing centers, but some vulnerabilities can cause at least great alarm. The biggest risks are often caused by vulnerabilities which can allow an intruder or abusive user to obtain privileged or elevated credentials. From the few vulnerabilities publicly submitted by supercomputer provider Cray [8], 57.1 percent relates to gaining privileges. The small number of submitted vulnerabilities can be explained by the fact, that the login nodes in systems delivered by Cray, are configured with software from third party providers, which provide their own vulnerability information.

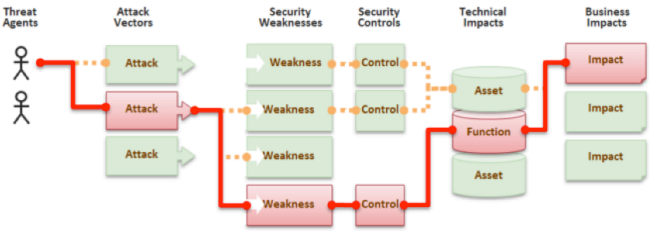

Figure 1. The OWASP Risk framework. CC-BY-SA by OWASP.

The vulnerability CVE-2017-5689 [9] is a good example of a vulnerability which can cause a distress at computing centers. According to the description, an unprivileged network attacker could gain system privileges to provisioned Intel manageability SKUs (Stock Keeping Units): Intel Active Management Technology (AMT), Intel Standard Manageability (ISM), and Intel Small Business Technology (SBT). As these technologies are widely used, there were many vulnerable systems, particularly if the systems were not protected by ab layered security architecture.

Another example vulnerability, CVE-2017-6074, caused alarm as almost all Linux platforms in use were affected while a local user could use the flaw to gain privileges on the system. Exploits were also available making attacks easy to almost anybody to try. Fortunately, the vulnerability could be fast mitigated with changes in kernel module configurations. Security patches became also soon available for all Linux distributions, but patches must be installed and for kernel patches systems also need to be restarted, which cause irritating downtime for users.

Our third example of a critical software vulnerability is CVE-2016-10030/ EGI-SVG-CVE-2016-1003 [10]. This application level buffer overflow vulnerability could allow an unauthenticated remote attack on the service, probably though an denial of service attack. The interesting part with this vulnerability was, that it had also been extensively assessed for a special context, the grid computing environment.

The EGI Software Vulnerability Group [11], [12] identifies and assess vulnerabilities for the EGI grid environment and require the affected EGI sites to patch the critical vulnerabilities at risk for to be excluded from the grid. EGI SVG also collaborate with other partners to identify vulnerabilities, and share information on vulnerabilities, encourage developers to write secure code, and encourage to

Develop awareness for security maintenance and secure configurations.

Another part of material for this paper, was an online survey sent in April-May 2017 to system administrators, security specialist and service managers in computing centers. The survey reached 51 respondents from 14 countries. The request to participate in the survey was sent to information email list for security collaboration among research computing centers and research infrastructures [13], [14].

The final explicit source of information for this paper was an interview [15], with very experienced system administrator at CSC – IT Center for Science, Mr. Esko Keränen, who used to be the lead administrator for CSC's Cray systems before his retirement.

III. RESULTS

The CVE based vulnerability statistics since 1999 show that risk based on software vulnerabilities are constant, the vulnerabilities shows a great variation on type and on severity.

Based on example vulnerabilities it seems that there is also a constant risk for severe and advanced attacks.

Our survey showed that most respondents, 73 percent saw that software vulnerabilities can cause severe risks for IT systems, 27 percent saw that major risk can be caused. Nobody saw that the risks are minor or not significant.

The risk caused by software vulnerabilities appears constantly according to 69 percent of the respondents while 27 percent saw that risks appear occasionally, when a major vulnerability has been exposed.

Answers about information on vulnerabilities were interesting:

- The Information was timely /late: 49 percent vs. 35 percent
- The Information was adequate/inadequate: 50 percent vs. 18 percent
- The Information was hard to identify /easy to identify: 41 percent vs. 38 percent
- The Information was biased: 22 percent

The three most important sources of information about software vulnerabilities were security advisories (87 percent) , CERT/CSIRT (71 percent) teams, and colleagues (53 percent). Blogs (16 percent), Twitter (24 percent) and other sites (33 percent) were least important source for this kind of information.

View on how big impact different software vulnerabilities have are shown in Figure 3, responses regarding patching and impact of systems compromise are shown in Figure 4 and in Figure 5.

An interesting result was also that the respondents saw that patching cause marginally (53 percent) or some downtime (47 downtime), nobody replied that downtime much downtime is caused by security patching.

Experience from our own sites, the datacenters of CSC – IT Center for Science Ltd. - show that security patching to mitigate software vulnerabilities causes clearly recognizable loss in availability, as the the monitoring of host sisu.csc.fi shows in Figure 1. Most of the breaks, except the breaks in January 2017 caused by issues with storage, were related were caused by a reboot required by a kernel level security patch

Sisu is a Massively Parallel Processor (MPP) supercomputer based on Cray XC40 technology serving researchers working in the Finnish universities, a typical example of a service for a national research computing center. It consists of nine high-density water-cooled cabinets for the compute nodes, and one cabinet for login and management nodes.

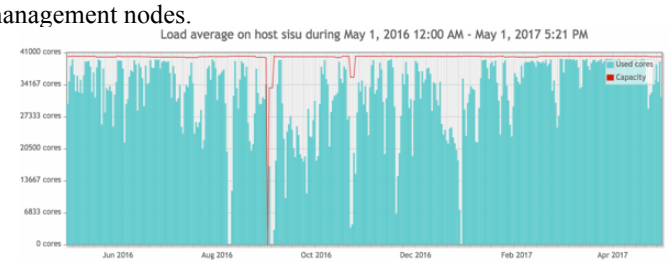

Figure 2. Load average and availability metrics for sisu.csc.fi

Reasons for breaks in system availability 2008-2011		
Reason for service break/ host	***Louhi (Cray)***	***Murska (HP)***
Security patch/ security incident	6.82%	12.98 %
Hardware	47.30%	61.35%
Software	18.11%	11.48%
IT infra (NFS & al.)	7.75%	0.14%
Data Center infra	0.35%	14.01%
Other	19,67%	0.04%

Table 1. Metrics and Best Practices for Host-based Access Control to Ensure System Integrity and Availability. [4]

Based on the survey, the interviews and other facts presented above, we recommend following measures to mitigate risks caused by software vulnerabilities:

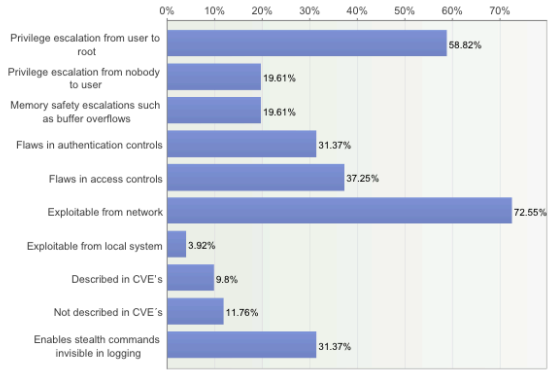

Figure 3.
The most dangerous software vulnerabilities (choose three).

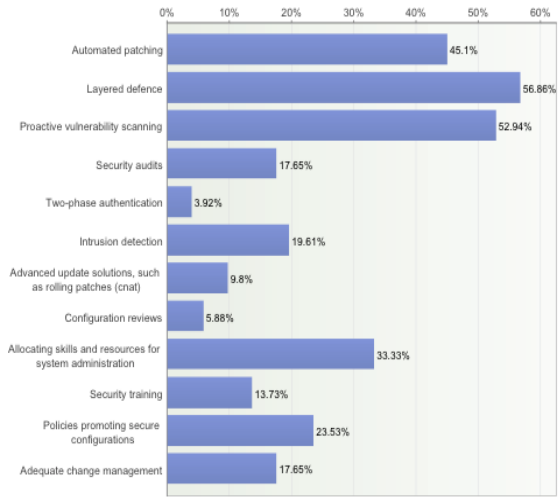

Figure 4.
The best way to mitigate risks cause by software vulnerabilities (choose three).

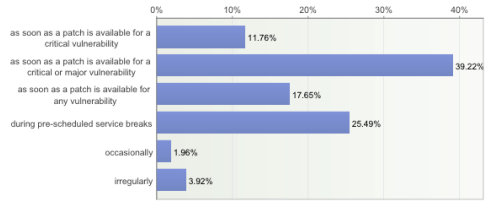

Figure 5.
When should patching of security vulnerabilities be done?

- Subscribe to platform specific vulnerability advisories, national cyber security agencies will seldom notify you about vulnerabilities for advanced systems
- Always react to all vulnerabilities, but take special care of those for which an exploit code is available (although it doesn't currently work, it can be improved fast)
- Beware of vulnerabilities hyped in common media, there will soon be exploit codes for these
- Odd processes, strange network traffic, suspicious software executions can be a sign of a system intrusion – although these are also quite normal phenomena in research computing
- You don't need to be root to install application software, limited privileges are sufficient – but the administrators do need root access to be able to do all admin tasks

IV. LIMITATIONS

The surveys made for this paper certainly has many limitations, too certain conclusions should not be made only based on the survey. Thus survey had a biased and a limited population. There was also some ambiguity in the questions and the context could sometimes also be understood in various ways.

No basic statistical analysis was made on based of the survey result because of limited number of responses. The survey just at best indicates views of a limited population

The responses also included some interesting comments

"The survey is not just about software vulnerability, but about security vulnerability in general. To me software vulnerabilities are in general caused by bugs within the software, which would limit the scope. Security vulnerabilities can be caused by many different reasons, system, kernel and software bugs, system and network issues, misconfigurations,…"

"In a large organization you have many types of devices… as well as the mixed nature of configuration it is a complex problem to keep up to date with vulnerabilities…."

"There is also a large number of people of varying skill level. The balance is very much in the favor of the attacker,…"

"Even with vulnerabilities in software there should be several other layers to mitigate that."

V. Discussion

Based on the findings above and by recommendations of Mr Keränen, here is a reminder to all sites about the profound and well known principles for good system administration and best security practices:

- Keep your system always patched
- Enforce adequate authentication and access controls
- Shut down unnecessary daemons
- Enable strict firewall rules,
- Ensure and check that you don't have any dormant test or service accounts enabled
- Use standard unix procedures to protect users from each others
- Access to other user's files must be restricted to administrators only
- Ensure that your system administrators have received adequate training and gained enough experience for secure administration practices – you must know what you do
- Apply strict access controls also - and specially – for administrators
- It is difficult to protect your system against bad system administration

Mitigating risks caused by software vulnerabilities at computing centers for research is a complex and dynamic tasks. Advanced and high risks can emerge on a short notice and it can be very difficult to obtain timely and adequate information about the vulnerabilities, which at worse could totally endanger the system integrity, if the vulnerability could be exploited to perform a full system compromise.

It is difficult to adequately identify anomalies indicating exploits in a very dynamic and complex computing environment, were the users constantly run their own experimental code.

Implementing best security practices nonetheless pays off and results in better efficiency and user satisfaction on the long run. A system intrusion would result in a major negative impact for the site. IT Systems and services should be secure by design.

Good ways to implement best security practices at research computing centers include:

- Sites should ensure that system administrators receive adequate training and skills development
- Implement layered defense
- Implement automated advanced patching – although patching cause some downtime
- Perform regular vulnerability scans
- Implement resilient host based and network based access controls
- Do explicit risk assessments and specify security requirements for your systems
- Do regular security compliance and vulnerability testing
- Include operational security (Change management, Incident management, Vulnerability management) in the IT service management of your site
- Designing a security architecture for your site
- Apply for a security certification
- Implement service monitoring

It is crucial that software and hardware providers have managed and mature processes for vulnerability management and efficient and effective methods to communicate security patches and related information to customers and to the user community.

Better technologies are also needed to cope with kernel level vulnerabilities.

Polite patch-and-reboot with fail active operational with solutions like kpatch [16] could be one way to limit downtime caused by applying security patches for linux kernel. This should probably also include a solution with some redundancy and job hibernation and restore from last known good state.

Acknowledgment

Special thanks for comment and advise to

- Mr. Esko Keränen, CSC iconic and retired Cray Administrator
- Dr. Linda Cornwall, STFC Rutherford Appleton Laboratory/ EGI Software Vulnerability Group

Many thanks for comments and advise from colleagues and friends at

- WISE - https://wise-community.org
- GÉANT SIG-ISM - https://www.geant.org/Innovation/SIG_TF/Pages/SIG-ISM.aspx
- CUG - https://cug.org/
- Funet Security Team
- CSC - https://www.csc.fi
- EUDAT – https://www.eudat.eu
- EGI - https://www.egi.eu/

References

[1] CWE Over Time. https://nvd.nist.gov/visualizations/cwe-over-time . Retrieved on 2017-01-20.

[2] Linux Kernel : Vulnerability Statistics. http://www.cvedetails.com/product/47/Linux-Linux-Kernel.html?vendor_id=33. Retrieved on 2017-01-20.

[3] W. Palm. Scaling Security in a Complex World. Proceedings of Cray User Group Conference (CUG 2016). 2016.

[4] U. Kaila, M. Passerini, and J. Virtanen. Metrics and Best Practices for Host-based Access Control to Ensure System Integrity and Availability. Proceedings of Cray User Group Conference (CUG 2011). 2011.

[5] NIST SPECIAL PUBLICATIONS (SP). http://csrc.nist.gov/publications/PubsSPs.html. Retrieved on 2017-01-20.

[6] Application Security Risks. https://www.owasp.org/index.php/Top_10_2010-Main. Retrieved on 2017-01-20.

[7] Top500 Sublist. https://www.top500.org/statistics/sublist/. Retrieved on 2017-01-20.

[8] Cray : Vulnerability Statistics. http://www.cvedetails.com/vendor/13/Cray.html. Retrieved on 2017-01-20.

[9] CVE-2017-5689. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5689. Retrieved on 2017-03-15.

[10] SVG:Advisory-SVG-CVE-2016-4303. https://wiki.egi.eu/wiki/SVG:Advisory-SVG-CVE-2016-4303. Retrieved on 2017-03-15.

[11] The EGI Software Vulnerability Group (SVG). https://wiki.egi.eu/wiki/SVG:SVG. Retrieved on 2017-03-15.

[12] Interview with Dr. Dr Linda Cornwall, chair of EGI Software Vulnerability Group. March 28, 2017.

[13] https://wise-community.org/

[14] https://www.geant.org/Innovation/SIG_TF/Pages/SIG-ISM.aspx

[15] Interview with Mr. Esko Keränen February 23, 2017.

[16] dynup/kpatch. https://github.com/dynup/kpatch. Retrieved on 2017-03-15.