CS 446/646 – Principles of Operating Systems

Abdur Rouf

December 2023

1. What is the responsibility of an OS vendor/distributor in the cases of where significant hardware flaws are discovered. What if their product is especially affected when compared to other competitors? What role do you believe an OS software engineer (OS developer, employee or manager) has to play in such a circumstance?

This question can be raised against the case discussed in section II (OS Security and Impacts on Performance or Economy). There have been discussed on popular two CVEs such as Meltdown and Spectre. These vulnerabilities are related to the processor's hardware level implementation flaws. Most of the processor has functionality to find the speculative instructions to run after the running current instruction. This feature facilitates to get the access to unauthorized process's data by exploiting the user privilege. The whole exploitation happened by creating race condition. After reported these vulnerabilities, all OS vendors released patches to fix this. However, the patches required reboot to the systems which impact high volume data traffic services. I've explored two more cases studies those are also related to hardware flaws which impacted users. The first literature I explored was Linux live device attack that facilitates the portable linux user some unauthorized access on host OS data [3]. This privilege leverages the malicious user by giving upper hand of changing the boot configuration of the host OS. Other case study was related to XBOX's secret boot block. It relies on this secret boot block embedded within a system ASIC ROM. However, this operation can be bypassed by some external ROM which acts as a decoy of the secret boot block [4]. This is hardware level weakness of XBOX boot design. Although, further software patches have been deployed, it hampered mass users.

These are the responsibilities in the case of the significant hardware flaws can be taken by vendors or distributors as per ACM code of ethics:

Communication with the users with transparency about the flaws: This action related to the section 1.3 (Be honest and trustworthy). According to the section, all parties related to the software have rights to know what the consequences of the flaws are those are exist. To acknowledge the code of ethics, the vendor's first responsibility is to communicate transparently with all related parties.

Patch development and deployment: This is the next tasks of the vendors to minimize the flaws at earliest possible time. This responsibility is also explained in section 2.9 which focuses on developing a patch for the vulnerable systems to minimize the harm. To achieve this code, vendors need to develop and deploy the patches as soon as possible.

Test compatibility: This step can be swapped with previous one. It is always necessary to make sure that deployed patches meet the requirements as well as it doesn't make another flaw. This responsibility is under the section of 3.1 of code of ethics.

Long-Term Support: After patches deployment, it should be carried out to later version so that the same vulnerability doesn't appear again. From ACM code of ethics discussed the same thing on section 2.9.

If the product is affected when compared with other competitors, these are measures those can be taken to reduce the harm.

Increase more resource: Sometimes more resources can be employed in specific issues to reduce the harm of the users as well as protect the reputation of the product. ACM code of ethics always encourage the software vendors to minimize the harm at any cost. And thus, employing more resources can be effective measure in this context.

Collaborate with other vendors: If the in-house resource insufficiency doesn't allow to do the above steps, collaborate with right parties or vendors sometimes help to remove the issues from the systems.

Additional Compensation: Some service might affect the user financially or affect the reputation of their business in B2B scenarios. In that case, some compensation can be applied to minimize user's loss.

As the software engineers or developers are the main workforce of a software vendor, they need to be pro active to minimize the harm. They can play role in some perspective such as vulnerabilities analysis before dive into the solution which is discussed in section 1.2 in ACM code ethics booklet. After analyzing the severity, they can develop patch and deploy to the systems by letting know all parties who are related with the harm. In a sentence, the OS developers core intention should be minimizing harm optimally by honoring confidentiality of the users. They should be fair to their actions so that particular users or related parties don't suffer to this.

2. What is the responsibility of an OS vendor/distributor when their platform is exploited (unspecified whether knowingly or not) to launch industry attacks that may geopardize more than just financial assets? What is the role of the OS engineer?

Aforementioned case study related to the disruption in centrifuges performing the Uranium enrichment process in Iran mentioned the importance of the question that should be encounter by incorporating the code of ethics which might reduce the harm of users in large scale. I also explored two more case studies those are also related to this question. First one is the SolarWinds Hack [1] and second one is the colonial pipeline ransomware attack [2]. Both the SolarWinds and Colonial Pipeline attacks relied on exploiting vulnerabilities in software supply chains to gain access to target systems, ultimately leading to major data breaches and disruptions. However, their consequences were same as aforementioned exploitation.

These are the responsibilities those should be carried out by OS vendors:

Blow the whistle: These types of attacks are mostly happened when higher officials or leaders are unaware about the attack. In this situation whoever finds the attack, he needs to blow the whistle to let everyone know about the severity of the attack. This action is under the section of 1.2 (Avoid harm).

Disclosure: Since a huge number of users might be affected from this, letting them about the vulnerabilities is another important responsibility that should be done by the OS vendor.

System Hardening and Security Updates: Vendors should continuously improve the security posture of their platform through system hardening techniques and regular security updates. Vendors need to ensure all kind of infrastructural elements that are necessary to meet this point.

Communicate with users: This kind of attacks or vulnerabilities make some financial or any other types losses to customers or users. Communicate with the user by giving compensation is must needed task that should be done by vendors.

Here are the responsibilities those should be come from OS development team:

Analyze vulnerability: Analyzing the vulnerabilities in quickest possible way is the next step of reporting the issues. Finding the actual root-cause helps to solve issues earlier.

Patch development and testing: Develop the patch, test on subset of the systems and deploy to the main system in earliest possible time is the another important responsibility which focuses on the reduce harm of the users that is discussed in section 1.2.

Patch deployment: After successfully tested all aspects of the new patches and related features, deploy to the main system in earliest possible time is the main job of developers.

Document the vulnerability: After releasing the patch, documenting the issues by explaining in plain words is another tasks that should be done by developers. Without proper documentation, further management will be difficult.

Proper communication with other stakeholders: After solving every problem, proper communication with corresponding stakeholders is the task for developers. In the code of ethics, every person in the organization is hired for specific competency. So, letting know the related stakeholders in timely manner is another important task.

3. What is the responsibility to an OS vendor/distributor with respect to safeguarding privacy? If your answer is situation-specific, elaborate. What is the role of the OS engineer? What concerns do you have regarding whether the OS and its safety features are proprietary or open-source?

This question can be encounter based on the case study discussed in aforementioned case study on OS security and legal questions (Section IV). There exists another dispute on whatsapp end to end message encryption policy [5]. In both cases, privacy policies communicated with the users were important and those are discussed in users data privacy section of ACM code of ethics booklet.

These are the responsibilities of OS vendor or distributor:

Protection of privacy: It is the vendor's or distributor's duty to protect user privacy and secure any personal information that is kept on their devices. This entails putting strong encryption techniques and security measures in place to shield user data from illegal access.

User control and privacy policy: The vendors should communicate with the users about their data privacy policy in clean manner. Which stakeholders have access to their data should be clear to users. Control over the data by users should be defined at the beginning.

Data deletion policy: How many days the users data will persist in the systems is another core point regarding safeguarding policy. After deletion of the data, how will be managed those data by vendor should be clearly communicated. There are two types of delete. One is soft delete and another is hard delete. Which deletion procedure has been implemented by the OS should be clear in safeguarding policy agreements.

Compliance with Legal Standards: Vendors that prioritize user privacy must also abide by the law when it comes to demands for encrypted data access from law enforcement. It is a difficult task that necessitates following legal requirements and due process to strike a balance between user privacy rights and law enforcement needs.

Now, we are discussing about the concerns with proprietary vs open source OS:

Rigorous community checking: When the code of the software is open source, it can be scrutinized easily by huge number of intellect brain that facilitates to reduce the potential harm of the systems as well as users become certain about the implementations and the agreements. On the other hand, In closed source systems, it's not possible to scrutinize the systems code against data privacy policy.

Transparency: Open source systems code are available to review any users which makes the user feeling transparent about their security and data privacy policy. In closed source system like IOS, it is not possible to verify by users. However, legal enforcements cross check the systems implementation against the policy. This cross checking is not enough in most of the time which leads the users less transparent about closed source systems.

Trade of between access and security: Whether the system is open source or closed source, there must be a certain boundary which allows the legal enforcement to access data when needed.

Quick patch update: Identify a vulnerability might be faster in open source systems but deploying the patch for the issues not easy going job when the system is open source. We know that open source systems update require consensus a significant amount of open source contributors. Patches have to go through some hierarchy that is less complicated in closed source systems like IOS or windows.

In conclusion, private systems frequently lack transparency even though they might provide a certain level of protection. Open-source systems encourage openness, but they could have trouble quickly fixing security flaws. For OS providers, preserving legal compliance, security, and privacy is a constant problem that calls for meticulous planning and research during the system's development and maintenance phases.

4. What do you believe are the ethics-related questions and principles that apply to an OS- development engineer, from the single-feature development roles, all the way to high-level management? Describe a form of integration between these levels that you believe is long- term sustainable.

Aforementioned last case study, The OS development process followed by Linus Torvalds introduces with ethics related quesitons as well as OS-development related work distribution and hierarchy within the development process.

Respectful to any level developers: In the professional code section, it has been mentioned that every persons who are related with development process should be well-treated. Acknowledging each persons contribution is one of the core practice of maintaining the code of ethics. Insulting behaviors or abuse of colleagues should be strictly discouraged and considered unacceptable.

Openness regarding decision making: ACM code of ethics teaches us to practice openness in volume of decision making. Sometimes, junior developer can propose better solution compared to senior's one. Practicing openness in decision can only help to find the absolute good decision. Decision only comes from leaders sometimes make the systems biased and prone to flaws. So, openness is must in decision making.

Accountability: In development process, every person related to OS development should have accountability on their tasks. Ability to accepting own fault can only help to solve the problem in earliest possible time.

Communication Skills: By developing proper communication skills with teammates helps to create leadership. In development, beside the technical excellency, good communication skills always give upper hand and make acceptable to other developers and leaders.

Consistent attitude: In team, every person should be fair to others. Biasing towards specific person makes the internal work environment unhealthy. So, showing consistent attitude with other persons is important.

From single-feature development roles to high-level management, a sustainable integration between various OS development levels can be accomplished by:

Respectful collaboration: According to the code of ethics, respectful collaboration with others is necessary to bring the organization towards success. Every software including OS requires a number of persons who constitute a team. If one person is disrespectful to others, it'll be hard for him to collaborate in team.

Transparent Governance Structure: Establishing a transparent governance structure enables proper decision making in the development process. It also encourages to acknowledge developers' contribution to the development process which works like a reward to their efforts.

Training: Provide training by mentorship helps developers to be better version with the days. Proper training should be available in every development team that accelerate team spirit and loyalty. Organization success significantly depends on the loyalty of their employees.

Proper planning and documentation: Continuous planning, execution and documentation can be helpful to become sustainable development process. This ensures a smoother transition during leadership changes and helps maintain ethical standards over time.

I believe integrating these practices in Linux or other big development process can create positive culture. ACM code of ethics always focuses on the good will of the bigger community. And this goal can only be achieved by being respectful to others, acknowledging contribution by incorporating proper governance, training the young minds etc. Though this list is not exhaustive. Only positivity can make an organization or team successful.

References

[1] Rahaf Alkhadra, Joud Abuzaid, Mariam AlShammari, and Nazeeruddin Mohammad. Solar winds hack: In-depth analysis and countermeasures. In *2021*

12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pages 1–7, 2021.

[2] Jack Beerman, David Berent, Zach Falter, and Suman Bhunia. A review of colonial pipeline ransomware attack. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, pages 8–15, 2023.

[3] M. junaid Gul, Riaz Rabia, Yaser Jararweh, M. Mazhar Rathore, and Anand Paul. Security flaws of operating system against live device attacks: A case study on live linux distribution device. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 154–159, 2019.

[4] Andrew Huang. Keeping secrets in hardware: The microsoft xboxtm case study. In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 213–227, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[5] S. Swetha. End-to-end encryption in messaging services and national security? case of whatsapp messenger. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, 06(14):Confcall, 2018.