COMP SCI 750 – Assignment 1
Gunnar Yonker

I. Access Control Models

1.

A. Bell-LaPadula Model (Ensuring Confidentiality):

The Bell-LaPadula Model follows these principles:

No Read Up (NRU) or Simple Security Condition – a subject with a certain clearance may not read data at a higher clearance level.

No Write Down (NWD) or *-Property – a subject with a certain clearance may not write to a data object at a lower clearance level.

Subjects/Documents	Doc1 (SECRET, {A,B})	Doc2 (TOP SECRET, {A,B,C})	Doc3 (UNCLASSIFIED, {B})
S1 (TOP SECRET, {A,C})	Read, Write	Read, Write	Read, Write
S2 (SECRET, {C})	Read	None	Read, Write
S3 (CONFIDENTIAL, {A,B})	None	None	Read, Write

S1: Can read and write all documents as the classification and categories allow it.

S2: Can only read Doc1 because of the matching category C. Can't read Doc2 because not all categories match and can read and write Doc3 due to lower classification.

S3: Can't access Doc1 or Doc2 due to higher classification levels, but can read and write Doc3.

B. Biba Model (Ensuring Integrity):

Biba is an integrity model and uses the principles:

No Write Up (NWU) – a subject may not write to an object at a higher integrity level.

No Read Down (NRD) – a subject may not read an object at a lower integrity level.

Subjects/Documents	Doc1 (SECRET, {A,B})	Doc2 (TOP SECRET, {A,B,C})	Doc3 (UNCLASSIFIED, {B})
S1 (TOP SECRET, {A,C})	Write	Write	None
S2 (SECRET, {C})	Read, Write	Write	Read
S3 (CONFIDENTIAL, {A,B})	Read, Write	Write	Read, Write

S1: Can write to all docs due to his high integrity but can't read Doc3 because of the NRD rule.

S2: Can read and write Doc1 and Doc3, can only write to Doc2 due to the NWU rule.

S3: Can read and write both Doc1 and Doc3 due to matching categories, can only write to Doc2 due to the NWU rule.

C. Combining Integrity and Confidentiality

Combining both models can become challenging because they might conflict in certain areas. For example, the Bell-LaPadula model restricts “write-down” while Biba restricts “write-up”. Lipner’s policy, which is an attempt to combine the confidentiality of Bell-LaPadula with the integrity of the Biba model, could be applied in such situations. It does this by introducing a trusted subject that has privileges to bypass certain rules (either read or write) when required. If this system can designate trusted subjects, it might effectively apply Lipner’s policy. However, if we were to merge the policies without introducing trusted subjects, there could be complications due to the conflicting nature of the two models. Using both models can result in a scenario where no data can be read or written because the read and write rules contradict each other.

Access Control Policy

Subjects/Documents	Doc1 (SECRET, {A,B})	Doc2 (TOP SECRET, {A,B,C})	Doc3 (UNCLASSIFIED, {B})
S1 (TOP SECRET, {A,C})	None	Read, Write	None
S2 (SECRET, {C})	Read	Write	None
S3 (CONFIDENTIAL, {A,B})	None	None	Read, Write

Since we need to implement both integrity and confidentiality controls, Lipner’s model would be a candidate to consider. Lipner’s model tried to address the scenario mentioned above which resulted in data being unable to be read or written to, by being more flexible. However, this means that it would require more specific implementation details and considerations on how conflicts between integrity and confidentiality controls are resolved.

2.

A. The Biba’s strict integrity policy follows the principle of “no read down, no write up”. This means that a user with a certain integrity level can only view content that is at or above their own integrity level and can only create content at or below their own integrity level. In the context of modern computers and diverse user needs, this model might be too restrictive for a few reasons.

i) Modern computing environments involve a wide range of operations, from software installations to internet browsing, where reading data from various integrity levels is common.

ii) The strict policy doesn’t reflect the practicalities of today’s diverse and interconnected digital landscape. Users often need to access and share data across different levels of trustworthiness.

iii) One of the cons of Biba’s integrity model indicates challenges with labeling for both subjects and objects on all computers. Many modern systems do not support this labeling.

iv) The Biba model doesn’t support network protocols that cater to such labeling, making it difficult to implement in today’s highly networked environments.

B. Low Watermark Policy: This policy is an extension to the Biba model that allows a subject to read everything. However, if they read data at a lower integrity level, their own integrity level is demoted to match that of the object they read. This makes the model more adaptable, but it doesn't necessarily ensure the integrity of objects. By allowing a subject to read down, their integrity can be degraded, which can potentially introduce risk when that subject interacts with other objects in the system.

Ring Policy: The Ring Policy allows any subject to read any object but only write down. This means that a subject can't corrupt a higher integrity object with lower integrity data. While this offers some level of protection against contamination of data, it allows all subjects to observe all objects. This means that while direct modifications might be safeguarded, indirect ones could take place. As a result, there is decreased integrity assurance.

C. Despite their limitations in ensuring total integrity, both the Low Watermark Policy and the Ring Policy are considered in some systems for the following reasons:

i) Flexibility – They offer more adaptability compared to the strict Biba policy, which is important in dynamic computing environments. The Ring Policy, for example, increases system flexibility by allowing any subject to observe any object.

ii) Usability – The strict enforcement of the original Biba policy might be too restrictive for many practical applications. By offering a middle ground, these policies make the system more usable without compromising integrity completely.

iii) Specific Use Cases – For certain environments or situations, the relaxed integrity controls offered by these policies might be deemed sufficient. For example, the Low Watermark Policy might be suitable for tasks like reading from a network socket.

iv) Building Block for Customized Solutions – By providing a range of policies with different levels of strictness, the Biba model allows system designers to choose the most appropriate policy for their specific requirements. They can even design hybrid solutions based on these policies.

In conclusion, while the Biba Integrity Model and its variations might not guarantee absolute integrity in all scenarios, they offer framework upon which more adaptable, practical, and usable systems can be designed.

3.

A. The use of Biba's integrity models is possible for the meat processing company, and here is how it could be done:

i) Factory Network – The industry control network in the meat processing factory will be the most sensitive, as it directly controls the production machinery. Disruption to this network could halt production and cause significant financial losses. Using Biba's strict integrity policy, machine operators, machine repairers, and managers in the factory can have high integrity levels, allowing them to modify settings and configurations.

ii) Intranet – The intranet is used for internal company communications. Employees from various departments would have different integrity levels based on their roles:

Purchase department, sales, order processors: Medium Integrity

CEOs and managers: High integrity to access sensitive company data.

IT and security: High integrity to maintain and monitor the system.

iii) Internet – Access to external communications, such as interacting with customers and suppliers. Given the potential risks from ransomware and other external threats, this network should have strict controls. Using Biba's Low Watermark or Ring Policy could work here. For example, if a salesperson accesses an external, potentially harmful source (like a phishing email), their integrity level drops to prevent them from contaminating higher-integrity systems.

iv) Remote Access (due to VPN): Employees working remotely should also be subjected to integrity checks. If they access lower integrity information from their home networks, their integrity levels should be adjusted accordingly when accessing company networks via VPN.

B. Remote working adds another layer of complexity to the security model. When employees connect to company networks via VPN, there's potential exposure to threats from their personal devices or home networks.

i) Separate Virtual Network – Create a separate virtual network for remote employees to connect to, which acts as an intermediary between the VPN and the main company networks. This "buffer zone" can have additional security checks.

ii) VPN Integrity Check – Before granting access to the main network, the VPN system should check the integrity level of the remote device. If a device accesses low integrity information (potentially harmful sources), the device's access to sensitive company data should be limited.

iii) Regular Security Audits – Ensure that remote devices are periodically checked for malware, and they have the latest security updates installed.

C. RBAC is a robust access control mechanism that grants access permissions based on roles within the organization. It offers a more structured and easily managed way of assigning permissions.

i) Roles Definition – Define roles for all employees:

Factory Operator, Factory Repairer, Factory Manager

Purchase Department, Sales, Order Processors

CEO, IT, and Security

ii) Permission Assignment – Each role has its associated permissions. For example:

Factory Operator: Modify machine settings, view machine logs

Purchase Department: Access to supplier databases, create purchase orders

iii) Hierarchy – Use Hierarchical RBAC for roles with inherent hierarchy. For example, a Factory Manager should have all the permissions of a Factory Operator (and more).

iv) Dynamic Role Activation – Use the concept of sessions to activate a subset of roles when required. For example, an IT employee might need temporary access to the factory network for troubleshooting. This can be achieved by activating a special session with the necessary permissions.

v) Benefits of RBAC –

Flexibility: easily add, modify, or revoke roles and permissions without impacting individual users.

Clear Audit Trail: Monitor actions based on roles, making it easier to track activities and potential anomalies.

Principle of Least Privilege: Assign only the necessary permissions for each role, reducing potential security risks.

Separation of Duties: Ensure no single role has excessive power or access, minimizing insider threat risks.

RBAC brings in flexibility and clarity, making it easier to manage permissions and roles within the company. Combining the two can provide a robust security framework against potential threats, including ransomware attacks.

II. Security policy for voting machine

1. In-person voting: The actual process of voting is kept secret through the use of private voting booths or stations where individuals can cast their votes without being observed. As for ensuring that the vote is received from a qualified voter, voters typically must present some form of identification or confirmation of their registration in order to vote. Commonly poll workers will have a list of registered voters for their polling station. Once the voter is identified and verified against the voter registration list they can vote. They will cross off names as individuals come in to vote, ensuring only those registered can vote and they can do it once. Voter secrecy is ensured because, while the system knows you've voted, it doesn't associate your identity with your specific vote.

Mail-in voting: Generally, mail-in voting works by sending out ballots to the registered addresses of qualified voters. The ballot typically comes with multiple envelopes; one to maintain the secrecy of the vote and another out envelope where the voter may have to sign or provide some other form of identification. This process ensures that only the intended recipient (a qualified voter) can vote while keeping their choice anonymous. The ballot itself, once separated from the identifying information, ensures secrecy.

2. Voting systems are designed to ensure a single vote by utilizing a registered voter list, and each voter's name is checked off as they cast their ballot. Once they've voted (either in person or by mail), this is noted in the system. If a person tried to vote at multiple stations, they would be detected as having already voted by the centralized voter system. Regarding computer systems, while it's possible for them to be hacked to allow a person to vote multiple times, the presence of backups, redundancy checks, and audits should ideally catch such inconsistencies. Voting systems are meant to be highly secure, but there is always a risk of being hacked.

COMP SCI 750 – Assignment 1
Gunnar Yonker

3. The primary flaw highlighted is the susceptibility of the voting machine to tampering, particularly the ease with which its vote-counting program can be replaced. If someone has even brief physical access, they can replace its vote-counting software using simple devices like memory cards, which poses a significant security risk. They can potentially be hacked or manipulated without leaving evidence of the tampering. Voting machines that do not produce a verifiable paper record can't be audited in a traditional manner.

4. For a malicious program to be effective, it would need to evade pre-election testing. This could be achieved by having the program act normally during such tests. The program could be designed to activate its malicious functions only under specific conditions, for example, after a certain number of votes have been cast or on the actual election day. By doing so, even if election officials test the machine before the start of voting, the malicious activities would not manifest. Another method could involve making a program that will self-delete or revert to a "clean" state are committing its malicious activities.

5. A sophisticated solution to prevent unauthorized software installations would be to implement a secure boot mechanism. This process ensures that the voting machine's operating system only boots up using software that has been digitally signed with a cryptographic key. In this setup, even if someone were to insert a malicious cartridge, the software wouldn't run unless it has the correct digital signature. The key would be securely held by election authorities and would be challenging to duplicate. The use of locks or tamper-evident seals on ports or access points could stop malicious installs from a USB drive. You could also disable the ability to install software unless it's from a verifiable source, along with having mechanisms to monitor and alert officials if unauthorized software is detected or if there is an attempt to load software.

6. Audits, especially manual ones on paper ballots, serve as a verification tool. When votes are manually recounted, the results can be matched against electronic counts to ensure there's no discrepancy. Even if only a sample of ballots are audited, this process can identify patterns of fraud or tampering. In case of a discrepancy, a wider audit can be triggered. Furthermore, the knowledge that such audits take place acts as a deterrent against tampering because it raises the chance of detection.