Questions On Computer Attacks

Q1. **What is the brain virus and how does it work?**

- The first computer virus, Brain, was discovered in 1986 and was created by two brothers, Basit and Amjad Farooq Alvi, who ran a computer store in Lahore, Pakistan.
- The primary purpose of the virus was not to cause harm, but rather to protect the brothers' medical software from being copied without their permission.
- Brain was a boot sector virus, which means it infected the boot sector of floppy disks.
- The boot sector is the area of a disk that is read by a computer's BIOS (Basic Input/Output System) when the computer is first started up.

Q2. **Brain Virus is a Self-replicating virus. Explain.**

- The virus was spread when users unknowingly infected their own systems by booting from an infected floppy disk.
- The virus was able to replicate itself and spread to other floppy disks, making it one of the first examples of a self-replicating virus.
- The virus replaces the original boot sector of a floppy disk with its own code, which is designed to check for the presence of the virus on the system it was infecting.
- If the virus was not present, it would copy itself to the boot sector and then infect any other floppy disks that were used on the infected computer.

Q3. **What is Internet Worm and its key characteristics?**

- An internet worm is a type of malicious software program that self-replicates and spreads across computers and networks without the need for human intervention. Unlike viruses, which typically require a host file to spread, worms can propagate independently by exploiting vulnerabilities in operating systems, network protocols, or software applications.
- Key Characteristics:
    1. Self-Replication: Worms can create copies of themselves and spread across networks, often rapidly and without user interaction.
    2. Exploitation of Vulnerabilities: They often exploit security flaws in software or network services to gain access to systems.
    3. Network Impact: Worms can cause significant disruptions by consuming bandwidth, overloading servers, and slowing down network performance.
    4. Payloads: Some worms carry additional malicious payloads, such as spyware, ransomware, or backdoors, which can further compromise affected systems.

Q4. **Prevention and Mitigation of Internet Worm**

- **Regular Software Updates**
    1. Patch Management: Regularly apply patches and updates released by software vendors to fix security vulnerabilities. Automated update tools can help ensure that systems are always up to date.
    2. Operating System Updates: Keep the operating system current with the latest security updates and service packs.
    3. Application Updates: Regularly update applications, including web browsers, plugins, and any other software that can be exploited by worms.
- **Firewalls and Intrusion Detection Systems**
    1. Network Firewalls: Implement firewalls to block unauthorized access to network resources. Firewalls can filter traffic based on predefined security rules.
    2. Host-based Firewalls: Install firewalls on individual devices to control incoming and outgoing traffic and protect against unauthorized access.
    3. Intrusion Detection and Prevention Systems (IDPS): Use IDPS to monitor network traffic for suspicious activity and automatically take action to prevent attacks.
- **Antivirus and Antimalware Programs**
    1. Real-time Protection: Use antivirus and antimalware programs with real-time protection to detect and block malicious software before it can execute.
    2. Regular Scans: Schedule regular scans of all devices to detect and remove any malicious software that may have bypassed initial defenses.
    3. Signature and Heuristic Analysis: Ensure that security software uses both signature-based and heuristic-based detection methods to identify known and unknown threats.

- **User Education**
    1. Phishing Awareness: Educate users about phishing attacks, which are often used to deliver worms. Train users to recognize and avoid suspicious emails, links, and attachments.
    2. Safe Browsing Practices: Encourage users to practice safe browsing habits, such as avoiding untrusted websites and not downloading files from unknown sources.
    3. Strong Password Policies: Implement strong password policies, including the use of complex passwords, regular password changes, and the avoidance of password reuse across multiple accounts.
- **Network Segmentation**
    1. Isolate Critical Systems: Segregate critical systems from the rest of the network to limit the spread of worms. Use VLANs or subnetting to create isolated network segments.
    2. Access Controls: Implement strict access controls to limit who can access different parts of the network. Use role-based access control (RBAC) to ensure that users only have access to the resources they need.
- **Backup and Recovery**
    1. Regular Backups: Perform regular backups of critical data and systems. Ensure that backups are stored securely and are not directly accessible from the main network.
    2. Backup Verification: Regularly test backup and recovery procedures to ensure that data can be restored quickly in the event of an infection.
- **Monitoring and Response**
    1. Continuous Monitoring: Continuously monitor network and system activity for signs of infection or unusual behaviour.
    2. Incident Response Plan: Develop and maintain an incident response plan to quickly contain and remediate infections.

Ensure that all staff are familiar with their roles and responsibilities in the event of an incident.

3. Threat Intelligence: Stay informed about the latest threats and vulnerabilities by subscribing to threat intelligence feeds and participating in security communities.

Q5. **Write a short note on Web Bug.**

- A Web bug is a hidden, transparent, but nonetheless "graphic" image that finds its way onto your computer. **They're small** "objects" that are embedded into a webpage or an email and are "activated" when you visit the site or open the email.
- Web bugs are placed on a website or in an email and they monitor, to a small degree, what you're doing while you visit the website or send the email.
- Web bugs are also referred to as "tags," ...tracking bugs," "pixel trackers" or "pixel GIFs." They're invisible because they're small, typically no larger than 1 pixel x 1 pixel.
- A Web bug gets your computer through an email, or it can be in a webpage that you visit. Some people might call it "spyware," in that it's used to take note of your online activity, but in most cases (especially with websites) it's not there to do any harm.

Q6. **What Web Bugs Do?**

- A Web bug gets your computer through an email, or it can be in a webpage that you visit. Some people might call it "spyware," in that it's used to take note of your online activity, but in most cases (especially with websites) it's not there to do any harm.
- Web bugs track in specific, purposeful ways some of your online behaviours when you receive an email or visit a specific website. Web bugs are custom-made: They are designed to monitor your activity (individually or as part of all website visitors) to give somebody helpful information.
- Website owners use Web bugs to know how many people visited their website. And ad networks (advertising companies focusing on the Web) use them to get customer behavior data. They'll use Web bugs in their ads to get an idea of how often an advertisement is appearing or being viewed. They can also use Web bugs to track an individual's response to online ads (one by one).
- With business emails (that are purely promotional), companies and online marketing agencies want to know if readers are opening the emails they receive. When the Web bug loads (which happens when the email is opened), the Web bug is embedded invisibly in the email graphics, so the company can find out if you opened the email, when you did it and so on.
    - It can gather the IP address of the computer
    - The URL of the web page the bug is located on
    - The URL of the page the bug came from
    - The time the bug was observed
    - A set cookie value
    - The type of browser that was used to get web bug graphic image
    - Going into your browser's settings to "turn off" (reject) cookies will stop Web bugs from tracking you.

Q7. **How to prevent or protect yourself against Web Bugs?**

- **Use Browser Extensions**
    1. Ad Blockers: Use tools that block ads and trackers.
    2. Privacy Extensions: Use tools that detect and block tracking technologies.
- **Email Security**
    1. Disable Image Loading: Configure your email client to not load images automatically.
    2. Secure Email Services: Choose email services that emphasize privacy and security.
- **Browser Privacy Settings**
    1. Block Third-Party Cookies: Disable third-party cookies in your browser settings.
    2. Do Not Track: Enable the “Do Not Track” feature in your browser.
- **Network-Level Protection**
    1. VPN: Use a Virtual Private Network to encrypt your internet traffic and hide your IP address.
    2. Privacy DNS: Use DNS services that block tracking domains.
- **Use Privacy-Focused Browsers**
    1. Privacy Features: Choose browsers with built-in tracking protection and privacy features.
- **Clear Cookies and Cache**
    1. Regularly Delete: Frequently clear cookies and browser cache to remove tracking data.
- **Monitor Permissions**
    1. Review Site Permissions: Regularly audit and manage permissions granted to websites.
    2. Disable JavaScript: Turn off JavaScript for websites that don’t need it.

Q8. Targeted malicious code- Trapdoors

- Targeted malicious code is specialized malware developed to compromise a specific individual, organization, or system. It is crafted to exploit distinct vulnerabilities and achieve particular objectives, such as unauthorized access, data theft, or surveillance, rather than indiscriminately affecting a broad range of users.
- A trapdoor is an undocumented entry point to a module. The trapdoor is inserted during code development, perhaps to test the module, to provide "hooks" by which to connect future modifications or enhancements or to allow access if the module should fail in the future. In addition to these legitimate uses, trapdoors can allow a programmer access to a program once it is placed in production.
- A trap door is a secret backdoor mechanism built into a system that allows an authorized person to access the system or specific functionality in a hidden way.
- It is often added to software or hardware systems by the system designer or developer, and is not disclosed to the end user or system administrator.
- The purpose of a trap door is to provide an emergency access mechanism that can be used by a system administrator to recover from a system failure or perform system maintenance activities.
- Trap doors can also be exploited by attackers to gain unauthorized access or control over a system.
- In some cases, a trap door may be intentionally added by an attacker during the development phase, enabling them to gain access to the system at a later time.

Q9. **Causes and Examples**

- **Causes**
- Developers usually remove trapdoors during program development, once their intended usefulness is spent. However, trapdoors can persist in production programs because the developers.
    - forget to remove them
    - intentionally leave them in the program for testing
    - intentionally leave them in the program for maintenance of the finished program
    - intentionally leave them in the program as a covert means of access to the component after it becomes an accepted part of a production system

- Examples

- 1980: In the 1980s, the US National Security Agency (NSA) was accused of including a trap door in the Unix operating system that was distributed to foreign governments. The trap door was reportedly designed to allow the US government to gain access to the systems of foreign governments that were using the Unix operating system.

- 2004: In 2004, a researcher discovered a trap door in the Diebold Election Systems voting machines used in the US. The trap door was designed to allow election officials to update the software on the machines, but it was not disclosed to the public or election officials, making it a potential target for attackers.

- 2013: In 2013, it was reported that the NSA had a program called Bullrun, which involved inserting trap doors into commercial encryption products, such as virtual private network (VPN) software, to enable the NSA to bypass the encryption and gain access to the encrypted data.

Q10. **Preventing Trapdoors**

- Trapdoors, or backdoors, are secret ways to access a system without authorization. Here are key steps to prevent them:
  1. Secure Development
     - Review Code: Regularly check code for hidden backdoors.
     - Use Analysis Tools: Scan code for vulnerabilities.
  2. Access Controls
     - Limit Access: Give users only the access they need.
     - Multi-Factor Authentication: Use MFA for critical systems.
     - Audit Logs: Track and review access to sensitive code.
  3. System Audits
     - Scan for Vulnerabilities: Regularly scan systems for weaknesses.
     - Penetration Testing: Simulate attacks to find backdoors.
  4. Network and System Security
     - Disable Unneeded Services: Turn off unnecessary features.
     - Use Firewalls and IDS: Monitor and control network traffic.
  5. Monitoring
     - Continuous Monitoring: Watch for unusual system activity.
     - File Integrity Checks: Detect unauthorised file changes.

Q11**. Explain Salami Attack:**

- A salami attack is a method of cybercrime that attackers or a hacker typically used to commit financial crimes.

- Cybercriminals steal money or resources from financial accounts on a system one at a time.

- This attack occurs when several minor attacks combine to create a sturdy attack. because of this sort of cybercrime, these attacks frequently go undetected.

- A salami attack is the theft of small amounts of money from a large number of accounts, often over a long period of time. It is named after the method of slicing thin slices of salami, as the thief is able to steal small amounts of money from many accounts without being noticed

Q12**. Explain the working of salami attack**

- During this kind of attack, an awfully insignificant change is introduced that goes completely unnoticed.
-
- As an example, the attacker inserts a program, into the bank's servers, that deducts a satiny low amount of cash from the account of each customer.
-
- No account holder will probably notice this unauthorized debit, but the attacker will make an outsized amount of cash each month.

Q13. State the different types of salami attacks:

- Salami Slicing:
    - Salami Slicing occurs when the attacker gets customer information, like debit/credit card details and other similar sort of detail by using an online database.

    - The attacker then deduct an awful touch of cash from each account and these amounts add up to an oversized amount of cash and this can be often invisible to detect such amount, since the amount is tiny

- Penny Shaving:
    - A penny-shaving attack is similar to a salami-slicing attack, but it involves the manipulation of financial transactions in order to steal small amounts of money from a single account over a long period of time.

    - The attacker infiltrates a company's financial system and begins making small, unauthorized changes to the amounts of financial transactions, such as rounding down the amount by a few cents or dollars and stealing a large amount of money from several bank accounts

Q14. **How to prevent yourself from the salami attack?**

- Users are encouraged to oversee their weekly transactions and month-to-month bank statements to shield their bank accounts from being hindered by a salami attack.

- You'll monitor any potential charges on your account by actively scanning through these activities.

- If you have got any issues with any strange charges on your account, contact your bank.

- Financial institutions, like banks, should also update their security so that the attacker doesn't become conversant in how the framework is meant.

- Banks should advise customers on the due to report any money deduction that they weren't tuned in to.

❖ **Presentation Link**

Web Bugs

A Web bug is a hidden, transparent, but nonetheless “graphic” image that finds its way onto your computer. **They’re small** “objects” that are embedded into a webpage or an email and are “activated” when you visit the site or open the email.

Web bugs are placed on a website or in an email and they monitor, to a small degree, what you’re doing while you visit the website or send the email.

Web bugs are also referred to as “tags,” …tracking bugs,” “pixel trackers” or “pixel GIFs.” They’re invisible because they’re small, typically no larger than 1 pixel x 1 pixel.

The Web bug shows up in a graphical/picture format called GIF (Graphic Interchange Format), common to the Web. To your browser, it looks just like any other picture or image on a webpage, and your browser doesn’t make a fuss about it. In other words, you and your Web browser simply don’t recognize bugs.

How do Web bugs get on your computer?

A Web bug gets your computer through an email, or it can be in a webpage that you visit. Some people might call it “spyware,” in that it’s used to take note of your online activity, but in most cases (especially with websites) it’s not there to do any harm.

What are Web bugs up to?

Web bugs track in specific, purposeful ways some of your online behaviours when you receive an email or visit a specific website. Web bugs are custom-made: They are designed to monitor your activity (individually or as part of all website visitors) to give somebody helpful information.

Website owners use Web bugs to know how many people visited their website. And ad networks (advertising companies focusing on the Web)

use them to get customer behavior data. They'll use Web bugs in their ads to get an idea of how often an advertisement is appearing or being viewed. They can also use Web bugs to track an individual's response to online ads (one by one).

With business emails (that are purely promotional), companies and online marketing agencies want to know if readers are opening the emails they receive. When the Web bug loads (which happens when the email is opened), the Web bug is embedded invisibly in the email graphics, so the company can find out if you opened the email, when you did it and so on.

- It can gather the IP address of the computer
- The URL of the web page the bug is located on
- The URL of the page the bug came from
- The time the bug was observed
- A set cookie value
- The type of browser that was used to get web bug graphic image

Going into your browser's settings to "turn off" (reject) cookies will stop Web bugs from tracking you.

How to Prevent or Protect Yourself from Web Bugs?

1. Use Browser Extensions

- **Ad Blockers**: Use tools that block ads and trackers.
- **Privacy Extensions**: Use tools that detect and block tracking technologies.

2. Email Security

- **Disable Image Loading**: Configure your email client to not load images automatically.
- **Secure Email Services**: Choose email services that emphasize privacy and security.

3. Browser Privacy Settings

- **Block Third-Party Cookies**: Disable third-party cookies in your browser settings.
- **Do Not Track**: Enable the “Do Not Track” feature in your browser.

4. Network-Level Protection

- **VPN**: Use a Virtual Private Network to encrypt your internet traffic and hide your IP address.
- **Privacy DNS**: Use DNS services that block tracking domains.

5. Use Privacy-Focused Browsers

- **Privacy Features**: Choose browsers with built-in tracking protection and privacy features.

6. Clear Cookies and Cache

- **Regularly Delete**: Frequently clear cookies and browser cache to remove tracking data.

7. Monitor Permissions

- **Review Site Permissions**: Regularly audit and manage permissions granted to websites.
- **Disable JavaScript**: Turn off JavaScript for websites that don’t need it.

TARGETED MALICIOUS CODE

Targeted malicious code is specialized malware developed to compromise a specific individual, organization, or system. It is crafted to exploit distinct vulnerabilities and achieve particular objectives, such as unauthorized access, data theft, or surveillance, rather than indiscriminately affecting a broad range of users.

Trapdoors

- A trapdoor is an undocumented entry point to a module. The trapdoor is inserted during code development, perhaps to test the module, to provide "hooks" by which to connect future modifications or enhancements or to allow access if the module should fail in the future. In addition to these legitimate uses, trapdoors can allow a programmer access to a program once it is placed in production.

- A trap door is a secret backdoor mechanism built into a system that allows an authorized person to access the system or specific functionality in a hidden way.

- It is often added to software or hardware systems by the system designer or developer, and is not disclosed to the end user or system administrator.

- The purpose of a trap door is to provide an emergency access mechanism that can be used by a system administrator to recover from a system failure or perform system maintenance activities.

- Trap doors can also be exploited by attackers to gain unauthorized access or control over a system.

- In some cases, a trap door may be intentionally added by an attacker during the development phase, enabling them to gain access to the system at a later time.

Causes of Trapdoors :

Developers usually remove trapdoors during program development, once their intended usefulness is spent. However, trapdoors can persist in production programs because the developers

- forget to remove them
- intentionally leave them in the program for testing
- intentionally leave them in the program for maintenance of the finished program
- intentionally leave them in the program as a covert means of access to the component after it becomes an accepted part of a production system

Trapdoors Examples (Backdoors)

- 1980: In the 1980s, the US National Security Agency (NSA) was accused of including a trap door in the Unix operating system that was distributed to foreign governments. The trap door was reportedly designed to allow the US government to gain access to the systems of foreign governments that were using the Unix operating system.

- 2004: In 2004, a researcher discovered a trap door in the Diebold Election Systems voting machines used in the US. The trap door was designed to allow election officials to update the software on the machines, but it was not disclosed to the public or election officials, making it a potential target for attackers.
- 2013: In 2013, it was reported that the NSA had a program called Bullrun, which involved inserting trap doors into commercial encryption products, such as virtual private network (VPN) software, to enable the NSA to bypass the encryption and gain access to the encrypted data.

Preventing Trapdoors (Backdoors)

Trapdoors, or backdoors, are secret ways to access a system without authorization. Here are key steps to prevent them:

1. Secure Development

- **Review Code**: Regularly check code for hidden backdoors.
- **Use Analysis Tools**: Scan code for vulnerabilities.

2. Access Controls

- **Limit Access**: Give users only the access they need.
- **Multi-Factor Authentication**: Use MFA for critical systems.
- **Audit Logs**: Track and review access to sensitive code.

3. System Audits

- **Scan for Vulnerabilities**: Regularly scan systems for weaknesses.
- **Penetration Testing**: Simulate attacks to find backdoors.

4. Network and System Security

- **Disable Unneeded Services**: Turn off unnecessary features.
- **Use Firewalls and IDS**: Monitor and control network traffic.

5. Monitoring

- **Continuous Monitoring**: Watch for unusual system activity.
- **File Integrity Checks**: Detect unauthorised file changes.

INTERNET WORM

An internet worm is a type of malicious software program that self-replicates and spreads across computers and networks without the need for human intervention. Unlike viruses, which typically require a host file to spread, worms can propagate independently by exploiting vulnerabilities in operating systems, network protocols, or software applications.

Key Characteristics of Internet Worms:

1. **Self-Replication: Worms can create copies of themselves and spread across networks, often rapidly and without user interaction.**

2. **Exploitation of Vulnerabilities: They often exploit security flaws in software or network services to gain access to systems.**

3. **Network Impact: Worms can cause significant disruptions by consuming bandwidth, overloading servers, and slowing down network performance.**

4. **Payloads: Some worms carry additional malicious payloads, such as spyware, ransomware, or backdoors, which can further compromise affected systems.**

Notable Examples: (text in orange will not be written in ppt)

- **Morris Worm (1988): One of the first and most well-known internet worms, which caused significant damage by exploiting vulnerabilities in Unix systems.**
- **ILOVEYOU Worm (2000): Spread through email and caused widespread disruption by overwriting files and sending copies of itself to contacts in the victim's address book.**
- **WannaCry (2017): A ransomware worm that spread through Windows systems by exploiting a vulnerability in the SMB protocol, encrypting files and demanding ransom payments.**

Prevention and Mitigation:

1. Regular Software Updates

- **Patch Management: Regularly apply patches and updates released by software vendors to fix security vulnerabilities. Automated update tools can help ensure that systems are always up to date.**
- **Operating System Updates: Keep the operating system current with the latest security updates and service packs.**

- **Application Updates: Regularly update applications, including web browsers, plugins, and any other software that can be exploited by worms.**

2. Firewalls and Intrusion Detection Systems

- **Network Firewalls: Implement firewalls to block unauthorized access to network resources. Firewalls can filter traffic based on predefined security rules.**
- **Host-based Firewalls: Install firewalls on individual devices to control incoming and outgoing traffic and protect against unauthorized access.**
- **Intrusion Detection and Prevention Systems (IDPS): Use IDPS to monitor network traffic for suspicious activity and automatically take action to prevent attacks.**

3. Antivirus and Antimalware Programs

- **Real-time Protection: Use antivirus and antimalware programs with real-time protection to detect and block malicious software before it can execute.**
- **Regular Scans: Schedule regular scans of all devices to detect and remove any malicious software that may have bypassed initial defenses.**
- **Signature and Heuristic Analysis: Ensure that security software uses both signature-based and heuristic-based detection methods to identify known and unknown threats.**

4. User Education

- **Phishing Awareness: Educate users about phishing attacks, which are often used to deliver worms. Train users to recognize and avoid suspicious emails, links, and attachments.**
- **Safe Browsing Practices: Encourage users to practice safe browsing habits, such as avoiding untrusted websites and not downloading files from unknown sources.**

- **Strong Password Policies: Implement strong password policies, including the use of complex passwords, regular password changes, and the avoidance of password reuse across multiple accounts.**

5. Network Segmentation

- **Isolate Critical Systems: Segregate critical systems from the rest of the network to limit the spread of worms. Use VLANs or subnetting to create isolated network segments.**
- **Access Controls: Implement strict access controls to limit who can access different parts of the network. Use role-based access control (RBAC) to ensure that users only have access to the resources they need.**

6. Backup and Recovery

- **Regular Backups: Perform regular backups of critical data and systems. Ensure that backups are stored securely and are not directly accessible from the main network.**
- **Backup Verification: Regularly test backup and recovery procedures to ensure that data can be restored quickly in the event of an infection.**

7. Monitoring and Response

- **Continuous Monitoring: Continuously monitor network and system activity for signs of infection or unusual behavior.**
- **Incident Response Plan: Develop and maintain an incident response plan to quickly contain and remediate infections. Ensure that all staff are familiar with their roles and responsibilities in the event of an incident.**
- **Threat Intelligence: Stay informed about the latest threats and vulnerabilities by subscribing to threat intelligence feeds and participating in security communities.**