Mid 2 Prep

Malware Types and their definition

1. **Virus:** attaches to a legit file and harms by replicating itself. User action required.
2. **Worm:** Self replicating, spreads through network exploiting vulnerabilities, no user intervention required.
3. **Trojan Horse:** seems legit, tricks users to install it then performs malicious actions like stealing data.
4. **Adware:** displays unwanted ads, alters user settings, redirects browser to generate ad revenue.
5. **Spyware:** secretly collects web history, keystrokes and other sensitive data for identity theft.
   e.g ↳ Keyloggers: downloaded w/ legit files
6. **Backdoor:** allows for unauthorized remote access to a system, control without detection e.g a backdoor tool "Back orifice"
7. **Ransomware:** encrypts victim data, locking him out of his own files ⇒ asks ransom for correction
8. **Root kit:** gains root access (admin privileges)
9. **Bootkit:** Malware attacks bootloader, runs before OS.

This was the list of malware. Now let us see how we can analyze these malware.

Analysis

1. **Static**: examination of malware without execution. reverse-engineering, decompiling, inspecting binary files. e.g IDA Pro
2. **Dynamic**: executing malware in a controlled environment. e.g cukoo Sandbox is a tool used for dynamic Analysis of Malware. use behaviour based detection tools.
3. **Behavioural**: focus on runtime behaviour and interaction w/ OS & network. We should use firewalls and intrusion detection Systems (IDS)
4. **Memory**: capturing and analyzing the memory of a running system to detect malware in RAM. e.g Volatility can be used to examine memory dumps or malware injected code. Implent frequent memory dumps and analysis on critical systems. Use DEP & ASLR
5. **Network Traffic**: checks and sees communication patterns, server connections. See unusual data transfer, malware related domains. Use firewalls and DPI, implement Anomaly detection.

Staele Caeris
ASLR

Vulnerability Types

1. Integer overflow: when an operation creates a value that exceeds the range. PHP overflow Attack (2012). Validate inputs and use appropriate data types.
2. String format: Attacker can input something which prints smth, runs arbitrary code according to will of the attacker. Use proper formatting, use safer snprintf.
3. Buffer overflow: when more data is written to a buffer than it can hold. Morris Worm. Stack canaries, ASLR, memory safe languages. Risk is high because it allows the attacker to write shellcode. Implement control flow integrity checks.

Database Security

4. SQL Injection Attack: Insert malicious SQL code into input fields. Heartland Payment Systems Breach (2008). Use prepared statements and parametrized queries to ensure input data is values, not code.

5. Database Access Control: restricts access to data based on who you are. Amazon's RBAC.

6. Database Inference Attack: inferred from accessible data and publicly available records. Healthcare database inference. Use differential privacy, add noise to the data

7. Database Encryption Methods: data is meaningless without decryption.

Web Security

1. Cross site request forgery (CSRF): attacker tricks a logged in user to do unwanted actions. Origin headers: reliable domains only, Synchronizer token Pattern (STP), referrer header: allowed websites only.

2. Reflected XSS: attacker makes the user click a legit link and malicious code is run in user's browser. Hijack session, steal data and cookies. Encoding & Validation

3. Stored XSS: malicious code is stored on the server and runs whenever server is accessed by other users.

4. DOM XSS: when client side of web scripts are modified in a harmful way. This opens the way for an attacker to inject harmful scripts.

Address Space Layout Randonization (ASLR)

We repeatedly relocate Memory blocks which are accessed for a certain task.