FIT2093 Tutorial 10

Topics

- Attacks on single devices
- Web-application hacking
- Network-based attacks

Review Questions

1. What is a *buffer overflow* and why can it be potentially used to attack a system? Explain in terms of abstract concepts (Input, Storage, Memory content, Memory addresses, etc.). What needs to be done at development time to prevent buffer overflow attacks? What can be done at run-time (or compile time).

   - Memory contains program code and data. If input is written into memory and is bigger than the space reserved for the data, it might overwrite other memory content, in particular memory content relevant for program execution, such as return addresses. Then, the program execution can change and the attacker can let the process execute commands that were not in the original program.
   - At development time, a program needs to include sufficient checks for input sanitization in order to prevent faulty (long) input to overwrite out of bounds memory.
   - At run-time (i.e. compile time) return addresses can be protected by so-called canaries that show that the particular memory segment was not changed. Also address randomization can make it much more difficult to jump to parts of the program or library that holds a function useful for the attack.

2. A current worm has distributed ransomware via a weakness in the SMB protocol on Windows computers. Explain why a properly configured firewall on a computer could have prevented the infection of this particular computer. SMB is a protocol for providing access to shared filesystems or other resources. Explain why a firewall-based protection might not have been possible for a server in an enterprise network.

   - SMB is using a particular port. If a firewall blocks this port, the service with the vulnerability is not accessible from remote. Thus, a worm cannot send data to this service and the vulnerability cannot be exploited.
   - A server might provide access to a filesystem using SMB. In this case, the SMB port (445 or 139) needs to be open to enable access. Blocking the port would result in denial of service.

3. Cookies can be used to identify a particular session between client and server. In combination with a TLS tunnel, cookies can provide a good solution for session identification, if the browser does not provide the cookie to another server, which it usually should not do. Why could an XSS attack still enable an attacker to take over the session. Read https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) to learn about other possible XSS Attack Consequences and go through the examples.

- The main effect of an XSS attack is that code coming from the attacker is executed on the victims machine as if it was coming from the correct server. Thus, this code gets access to the session cookie and can just forward cookie content to the attacker.
- Examples on the OWASP side are really useful to get an understanding how and why XSS actually works.