Vulnerabilities

eros

April 8, 2021

1 Bugs vs Vulnerabilities

Bug Problem in the execution of the software that leads to unexpected behaviour

- **Replicable** In the same conditions, the same unexpected behaviour manifests
- **Type** Logic, configuration, design, implementation
- **Fix priority** Rank how severe the bug is
- **vs Feature** If it's documented, it's a feature

Vulnerability A bug that results in a security breach or a violation of the system's security policy

Let's assume that all the functions beign called in the pseudo-code below are error-free.

```
    gets(password);
    correct_pwd=lookup(username,database);
    if(correct_pwd!=password)
        printf('Login failed');
        return err;
    else{
        printf('login succeded');
        exec(context);
    }
    return x;
}
```

Problem Missing parenthesis in the `if` statement

Bug No matter the correctness of the password, an error is always returned

Assuming all the function are error-free, what bug is affecting the code below?

```
    gets(password);
    correct_pwd=lookup(username,database);
    if(correct_pwd=password)
        printf('login succeded');
        exec(context);
    else{
        printf('Login failed');
        return err;
    }
    return x;
}
```

Problem Missing parenthesis in the `if` statement

Vulnerability No matter the correctness of the password, the login succeeds

2 An overview

Types of vulnerabilities

Configuration vulnerabilities Software/system is wrongly configured (e.g. accept SSH root connections from any IP)

Infrastructure vulnerabilities Design/implementation problems affects security of a system (e.g. sensitive database in a network's DMZ)

Software vulnerabilities Design/implementation of a software module can be exploited to bypass security policy (eg. bypass authorization mechanism)

Who is responsible for a vulnerability?

System owner Configuration and infrastructure vulnerabilities

Software vendor Software vulnerability

The increasing complexity of software leads to an evergrowing amount of vulnerabilities

- High quality standard code has statistically 4-6 bugs every 1000 lines of code
- The amount of discovered vulnerabilities is a small fraction of the total
- Of the discovered vulneravilities, only some are publicly disclosed

3 Vulnerability discovery

Vulnerabilities may be discovered

Internally Thanks to quality and assurance (Q&A) process. The vulnerability is patched and the customer must be informed.

Externally An external security researcher. Today, software vendors are liable for their software vulnerabilities and must fix them. This created a market, where companies put up bounties for vulnerabilities and white hat hackers receive prices

3.1 Tools

The standard, dull way to search for vulnerabilities is via

Vulnerability scanning Check and report using a knowledge base of vulnerabilities with no exploitation (e.g. OpenVAS, Matasploit).

Black box Search as an external entity with no privileged knowledge

Grey box Search as an internal or partially internal user. In this case non-disclosure agreement (NDA) must be signed by the parts (tester and target).

Penetration testing Given a list of vulnerabilities, test their exploitability without concluding the attack at the very end (e.g. Matasploit). A NDA is always required.

When running these tools, the real difference is made by who is actually running them: the expertise of the tester can enhance the baseline of the tool. Whoever, given the large amount possible vulnerabilities, a team of specialized experts is usually required to discover the most insidious vulnerabilities.

3.2 Human expertise

A vertical, deep, low-level understanding of the system/software design is often required. Uncommon skills may be required to find many vulnerabilities.

3.3 Techniques

Code lookups Requires the source code and a codebase for known patterns

Fuzzing Try to crash the program with repeated semi-automatic random input (long and time consuming)

"Google hacking" Search in the web for vulnerable code and target the resulting pages

4 Vulnerability handling (ISO 30111)

Acceptance Of vulnerability discovery from internal/external sources by providing entry points

Verification Given a discovery, its investigation goes by steps

- Is it a vulnerability?
- Is the vulnerability affecting a supported version of some software the vendor maintains?
- Is the vulnerability exploitable?
- What is the root cause? Are there similar vulnerabilities?
- Whats is the potential threat of this vulnerability?

Resolution Given a verified vulnerability, decide how to resolve it. In any case, resolutions must be tested before any release

Configuration vulnerabilities Simply provide guidelines/advises for a correct configuration

Code vulnerabilities Provide a patch, timing depends on release schedule

Critical vulnerabilities Provide mitigations before patch is released

Release Resolutions can be released via different channels

Push to web-services Customer subscribes to the service and receives the patch

Pull stand-alone product Customer has to actively reach the new product version

Post-release Most provide customer support

5 Vulnerability disclosure (ISO 29147)

Defines information exchange between vulnerability finders, vendors and possible coordinators. Related to de-facto standards include STIX and TAXII.

6 Confidentiality

Vulnerability information

- Is considered sensitive and confidential by vendors
- Should travel only through a secure communication channel
- Depending on the vendors policies, information about the vulnerability may be (or not) published, if so always after a patch release.

The reward system can be tricky since

- External finders have to communicate only the right amount of info to get rewarded
- Agreement between researcher and vendor must be protected. Often requires third party mediators that
  - Hold the vulnerability info for a certain amount of time (60-90 days)
  - After this period, the vulnerability is disclosed
- What is a fair reward? Money? Credits?
- A proof-of-concept is required to proove the exploitability (that is not published)

Zero day vulnerabilities are

- Disclosed before patch release
- The target of the most effective and interesting attacks
- The interest of the Google Zero Day Project

7 Disclosed vulnerabilties

Public database National Vulnerability Database (NDV), a public NIST-maintained database of discosed vulnerabilities. Each entry has a Common Platform Enumeration (CPE), a list of systems affected by the vulnerability (provided by vendors, not always reliable)

Private feeds Specialized, paid services that release weekly/monthly information

Several communities and initiatives tried to provide classification of vulnerabilities

- Open Web Application Security Project (OWASP)
- Common Weakness Enumeration (CWE)
- Computer Security Incident Response Team (CSIRT) can be used to report vulnerabilities if the organization doesn't provide an access point

8 ACM Code of Ethics and Professional Conduct