

Internet infrastructure

Prof. dr. ir. André Mariën

Topic

Vulnerability and patch management

Requirements

- Security principle:
 - Everything can and will fail
- Consequence:
 - Prepare for failure:
 - No single point of failure
 - Fail closed
 - Recovery strategy
- Typical failures
 - Failures: Human error, hardware failure
 - Malicious: hackers

Hacking:

- Find vulnerability
 - Systematic search
 - Known weakness against all interesting targets
 - Vulnerability = possible problem
- Create exploit
 - Demonstrate how the vulnerability can be abused
 - Provides certain functionality
 - Crash the service or system, read data from the application or system, modify data on the application of system , remote shell
- Use exploit
 - Actual attack on third party system = breach of the law(s)
- Create worm
 - Self replicating exploit
 - Identify other targets, attack, propagate, ...
- Unleash worm
 - Infect some systems with the worm

Defense against hacks

- Retroactive working! Always in defense mode
 - The attacker has the advantage
- One option: Own quality/security: look for vulnerabilities
 - Specific testing (fuzzing for instance)
 - Customer bug reports
- Main trigger: a vulnerability report

From vulnerable to safe

- Vulnerability reported
 - Vulnerability tracking
 - Relevance for me?
 - Need for sufficiently detailed asset inventory (asset management)
 - Include version, but also optional packages
- Assess risk
 - If an exploit would appear, what could be done?
 - How easy would it be to develop an exploit, and to exploit it effectively?
 - Some third party providers exist that deliver this information
 - But need to assess in own context

(cont)

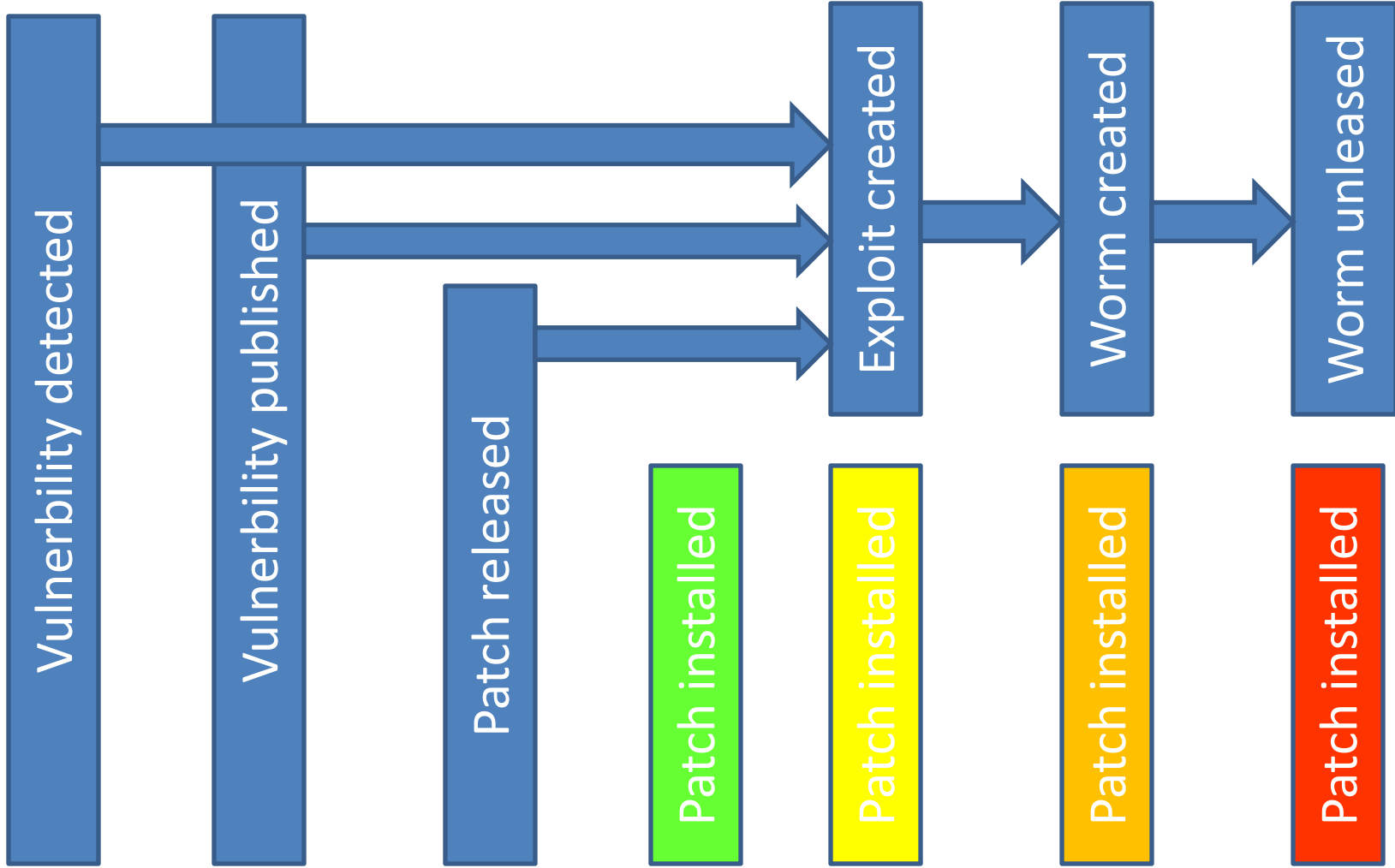
- Patch tracking
 - Push or pull
- Install patch
 - Apply automatic (desktop) or not (servers)
 - Patches may break other (sub)systems
 - Need to go through test, stage, production
- Exploit creation
 - How to really abuse a vulnerability
 - Similar to previous, all new?
- Patch analysis
 - Delta analysis shows details about fix, and thus problem
- Worm potential?
 - Might decide to create worm

Race condition

- Patch installation comes first: you win
- Exploit comes first: you loose
- Worm comes first: big trouble

- If the vulnerability is detected by black hat, the exploit may exist before the security community knows about it, we have a zero-day condition
 - If it is a worm, a very bad one

Race



Some processes

- Vulnerability tracking
 - Track
 - Assess
- Asset management
 - Necessary supporting process
 - May be supported with discovery tools (OS, services fingerprinting)
 - May drive vulnerability tracking
- Patch management
 - Track
 - Assess
 - Install
 - Auto update may not be the best
 - Tools: PatchLink, Microsoft's Software Update Service, citadel, ...

References

- Creating a Patch and Vulnerability Management Program, NIST Special Publication 800-40