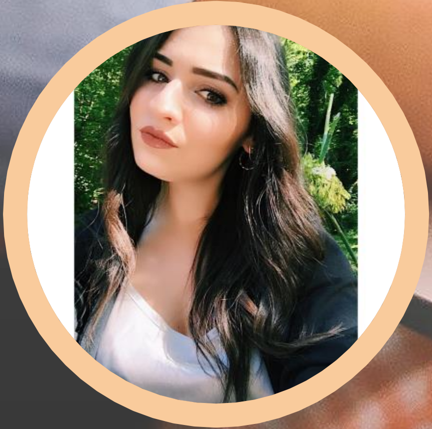
Software Vulnerability: Life-Cycle



Our Team Content

Life-cycle

1

Discovery

Is when it is discovered by the vendor, a hacker, or any third-party software analyst.

2

Disclosure

Starts with the public disclosure of the vulnerability

3

Exploitation

When the vulnerability is exploited by hackers.

4

Patching

When the vendor patch the vulnerability.

Why is important?

Vulnerability lyfe-cycle

Is helpful for...

Deployment
Of best practices in the software development processes.

Insights
About the previous security incidents that are helpful in their audit.

Security policies
Can handle future attacks and threats more effectively.

Helps customers
To assess the security risks associated with the software products.

Terminology

1 **Vendor**

Develops a software product and is responsible to keep it secure.

2 **Hacker**

Releases exploits for the vulnerabilities in the software products.

3 **Independent organization**

Discovers and discloses vulnerabilities but is not involved in the development of patches or exploits.

4 **Disclosure date**

Date when information about vulnerability is made publicly available.

5 **Patch date**

Date when a vendor provides a solution.

6 **Exploit date**

Date when a vulnerability is exploited.

7

Exploit - Disclosure

The duration between the exploit date and the disclosure date.

8

Patch - Disclosure

The duration between the patch date and the disclosure date.

9

Patch - Exploit

The duration between the patch date and the exploit date.

10

Access Vector

Indicates if local or network access is required to exploit the vulnerability..

11

Access complexity

Measures the complexity of the attack required to exploit the vulnerablity..

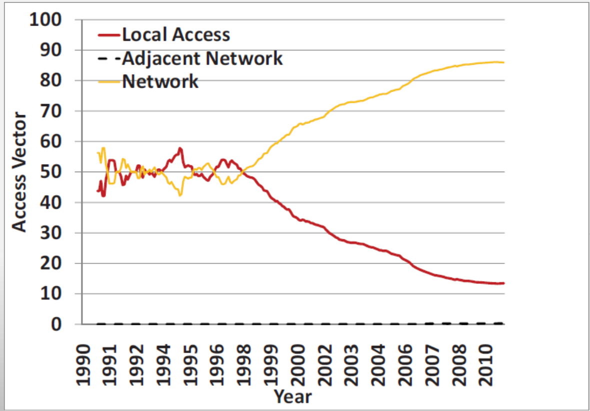
12

Integrity impact

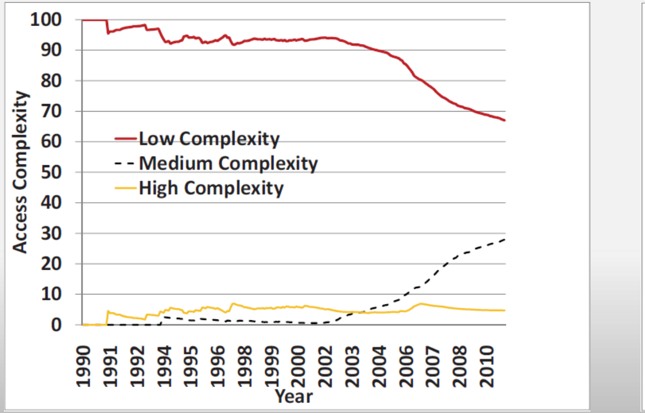
Measures the potential impact on the integrity of the system.

Vulnerability over time

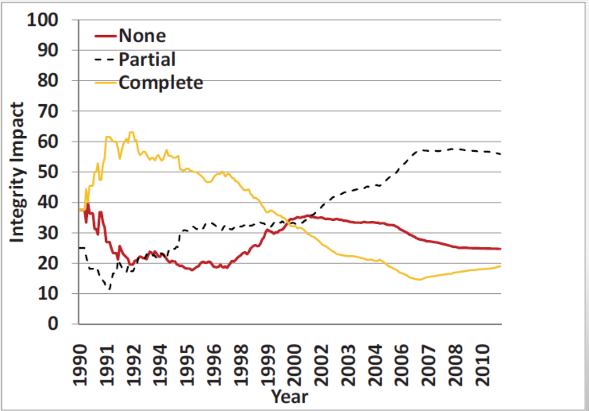
Access vector evolution

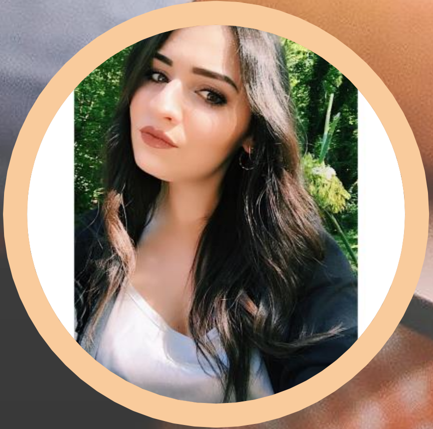


Access complexity evolution



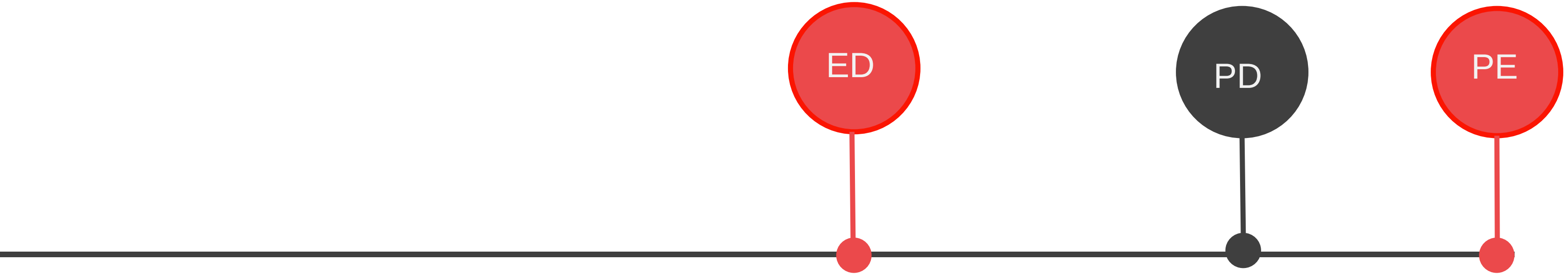
Integrity impact evolution





Our Team Content

Division of Dataset



ED

15.456

PD

9.667

PE

1.424

Exploitation behavior
&
Patching behavior

Bad side

case:

1. $t_{ed} < 0$

1. $t_{ed} = 0$

1. $t_{ed} > 0$

Good side

case:

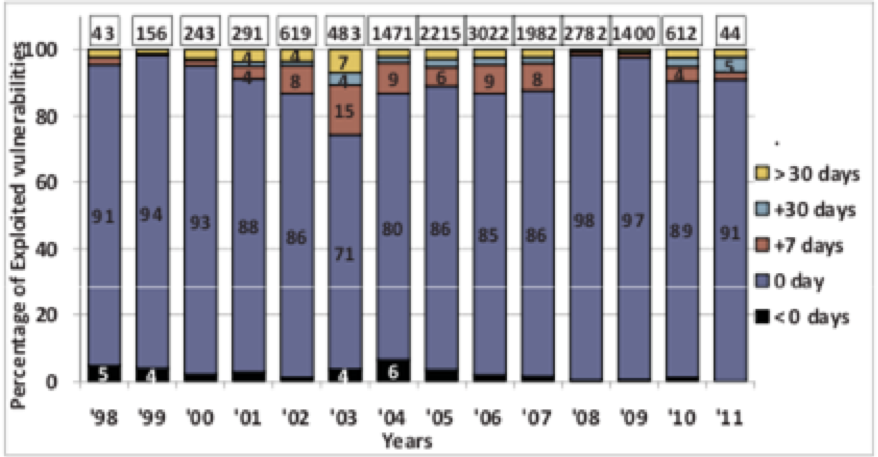
1. $t_{pd} < 0$

1. $t_{pd} = 0$

1. $t_{pd} > 0$

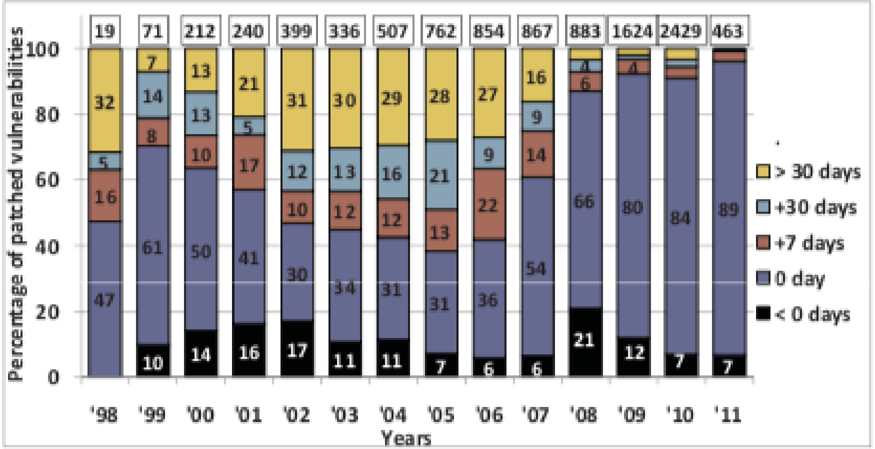
Exploitation behavior
&
Patching behavior

Bad side



The change of exploitations over the years

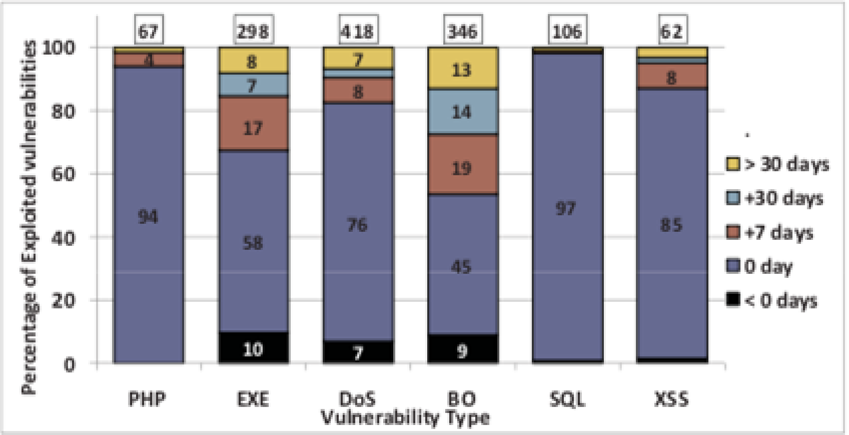
Good side



The change of patches over the years

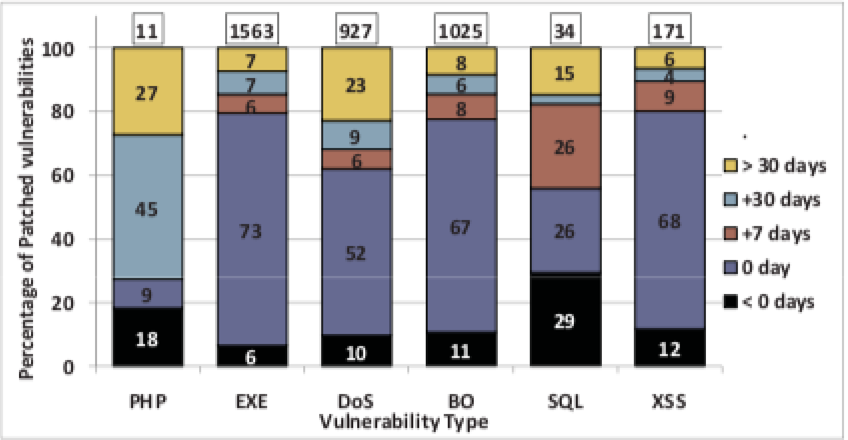
Exploitation behavior
&
Patching behavior

Bad side



Exploitation trend in clusters

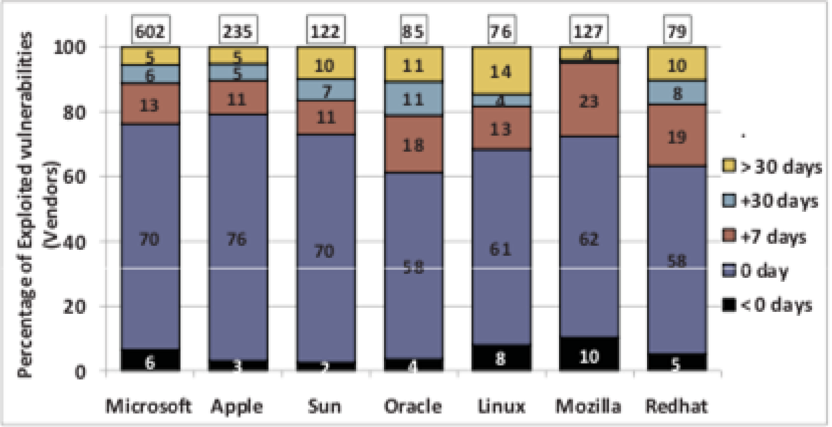
Good side



Patching trend in clusters

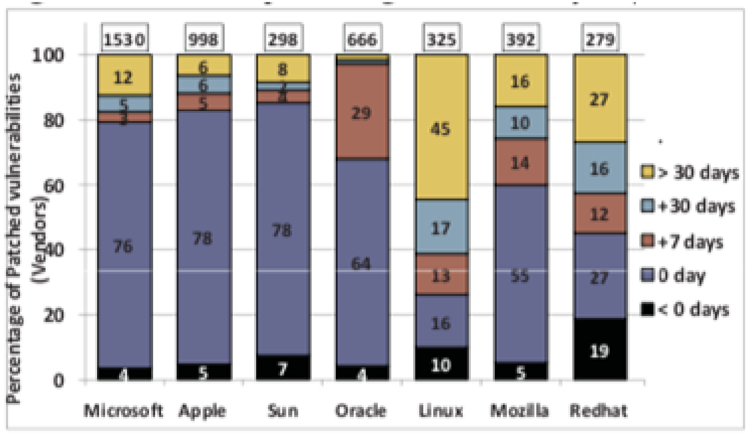
Exploitation behavior
&
Patching behavior

Bad side



Exploited vulnerabilities for vendors relative to disclosure dates

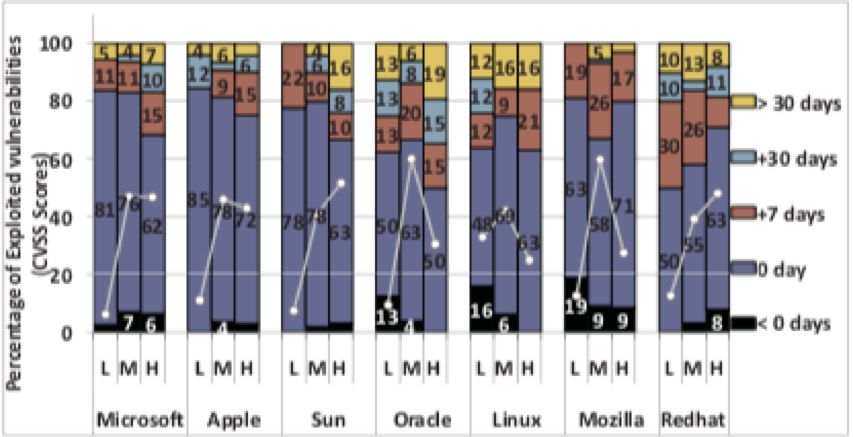
Good side



Exploited vulnerabilities for vendors relative to disclosure dates

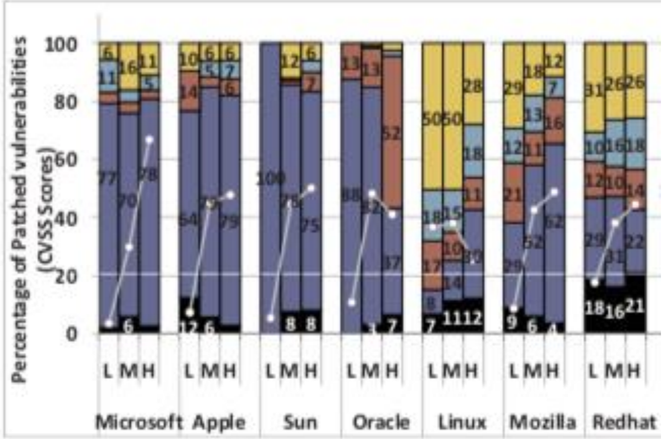
Exploitation behavior
&
Patching behavior

Bad side



Exploited vulnerabilities for different CVSS scores

Good side



Patched vulnerabilities for different CVSS scores

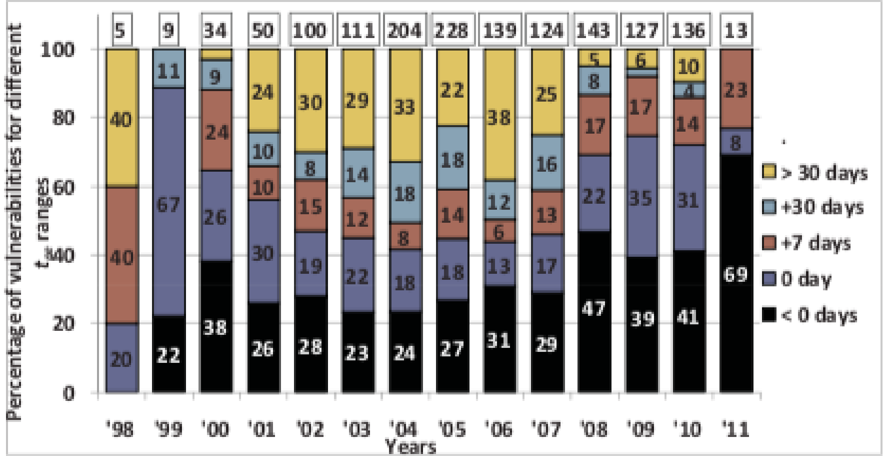
Patching vs Exploitation

case:

1. $t_{pe} < 0$

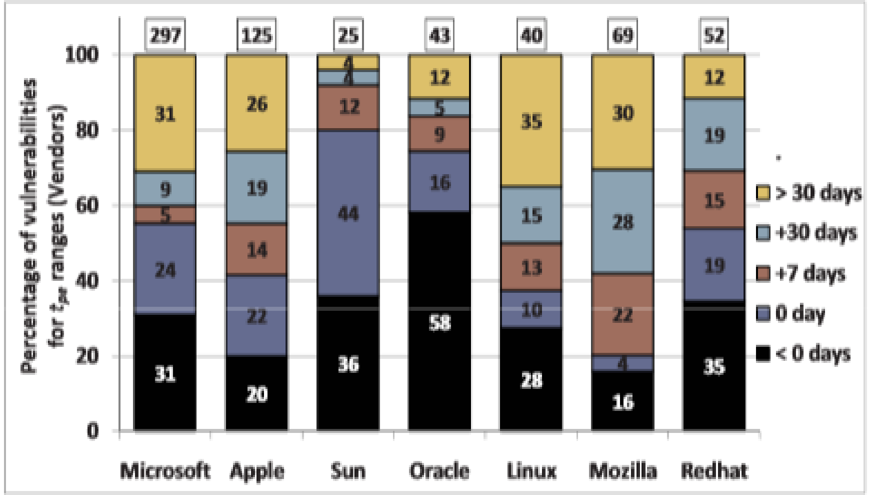
1. $t_{pe} = 0$

1. $t_{pe} > 0$

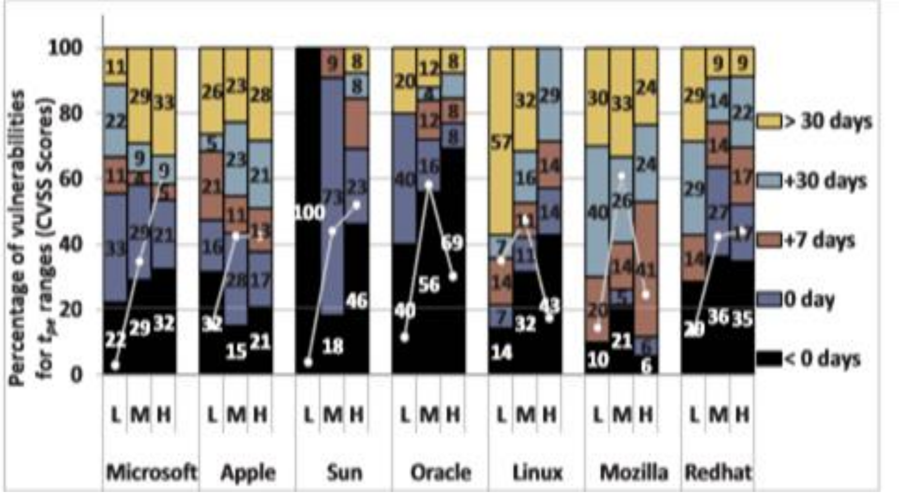


Yearly change in patching vs exploitation trends

Patching vs Exploitation



Patched vulnerabilities for vendors relative to exploit dates



Patched vulns. relative to exploited vulns.

Conclusion

1 Since 2008, the vendors have been becoming more agile in patching the vulnerabilities, and the complexity of vulnerabilities has been increasing.

2 The percentage of remotely exploitable vulnerabilities has gradually increased to over 80% of all the vulnerabilities.

3 Most exploited form of vulnerabilities are DoS, BO, EXE.

4 Patching of vulnerabilities in closed-source software is faster compared to open-source software and at the same time the exploitation is slower

Thank you