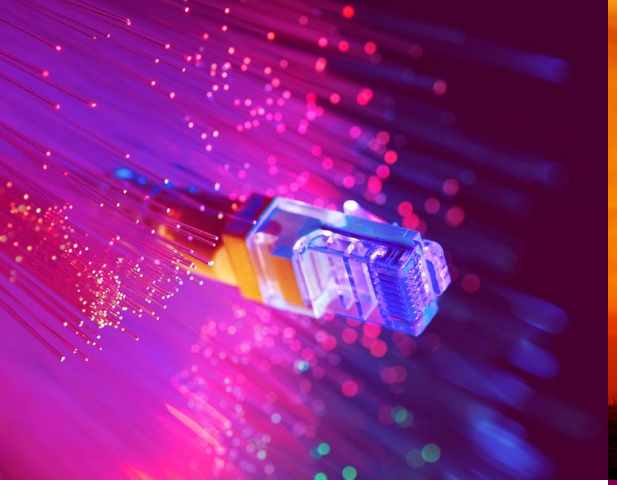


CSC

Advanced Risk Mitigation of Software Vulnerabilities at Research Computing Centers

Cray User Group Conference; May 11, 2017, Seattle

Urpo Kaila <urpo.kaila@csc.fi>

CSC – Finnish expertise in ICT for research, education, culture and public administration

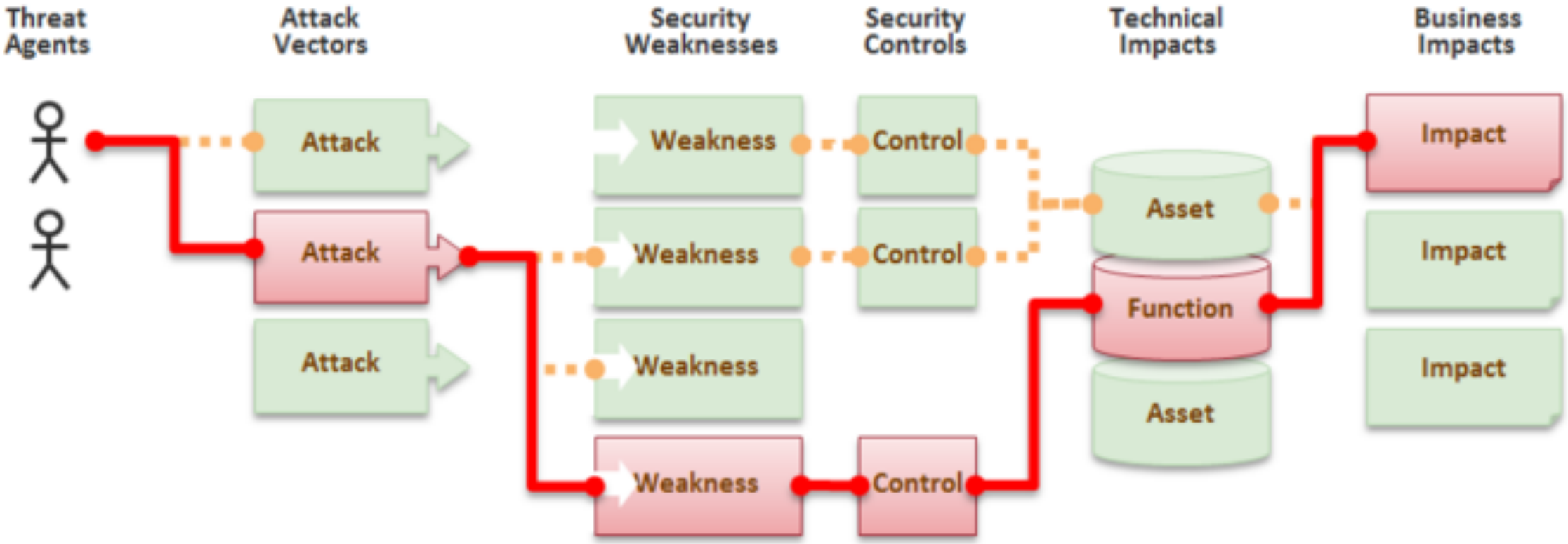
Outline

- Terms and concepts
- Examples of vulnerabilities
- The dynamics of information about vulnerabilities
- Security patching vs availability
- Results from a survey
- Conclusions
- Suggestion on how to improve vulnerability management

Terms and concepts

- Software vulnerabilities
  - *Weaknesses in a system's software design which can be exploited*
  - *Other vulnerabilities: insecure configurations, flaws in operational, network, or physical security, legal issues*
- Risk management
  - Identifying and mitigating threats (software vulnerabilities can be one of them)
  - Scopes for risk management: strategic risks, operational risks, damage risks
  - Standard risks/Advanced risks: probing for weak accounts/Advance persistent threats
- Information security
  - Information security is about protecting assets (systems, data, services and reputation) against risks with security controls
    - Security controls: technical/managerial; reactive/proactive; detective/preventive/reactive; compliance
    - Preservation of confidentiality, integrity and availability of information

The OWASP* Risk framework

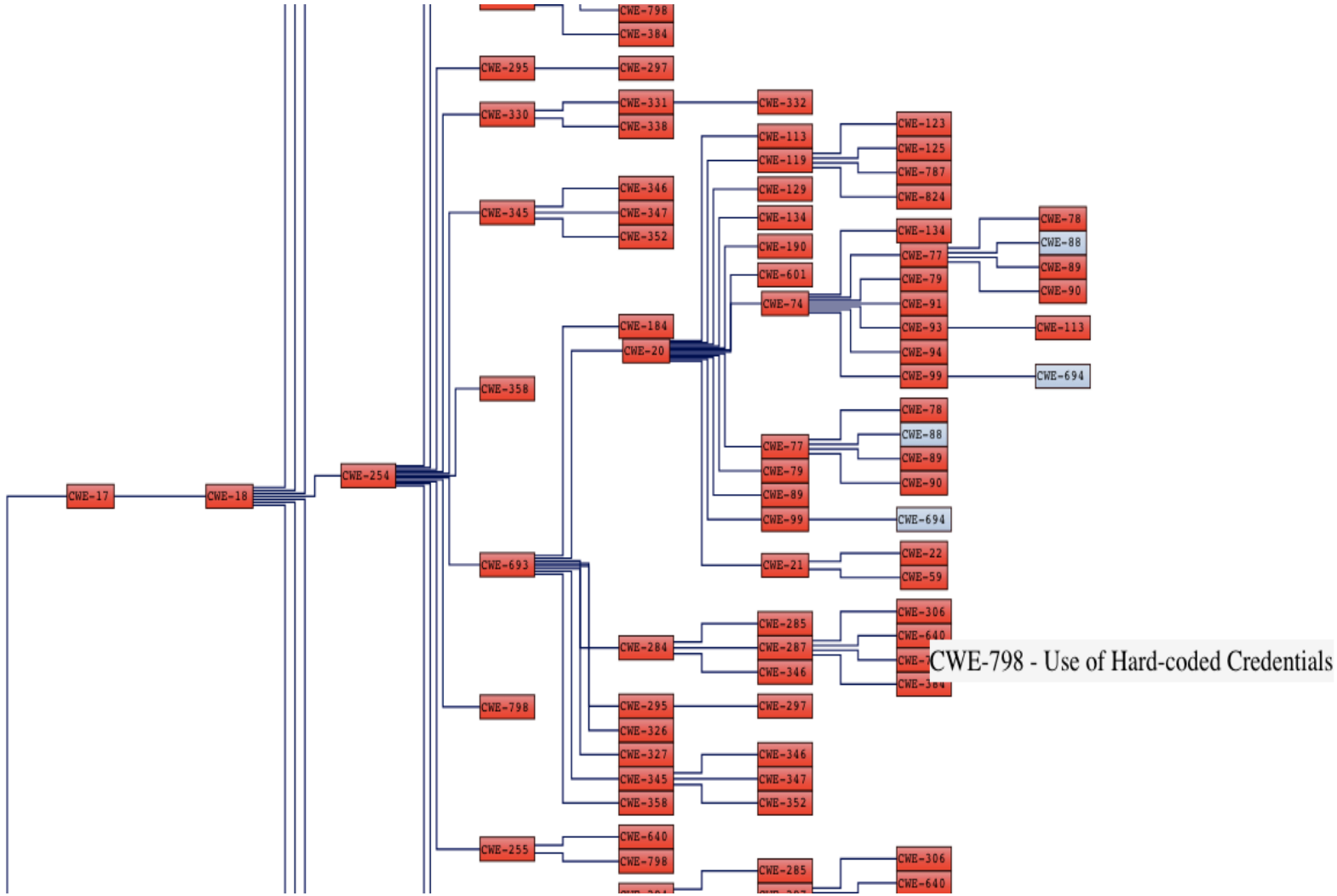


* https://www.owasp.org/index.php/Top_10_2010-Main

CVE - The Standard for Information Security Vulnerability Names

- Common Vulnerabilities and Exposures (CVE®) - Publicly known cybersecurity vulnerabilities

- The Common Weakness Enumeration Specification (CWE) provides a common language for software security vulnerabilities
  - https://nvd.nist.gov/vuln/categories



Example vulnerability: CVE-2017-5689

Description:

An unprivileged network attacker could gain system privileges to provisioned Intel manageability SKU*s: Intel Active Management Technology (AMT) and Intel Standard Manageability ….

References:

http://www.securityfocus.com/bid/98269

Bugtraq ID: 98269

Remote: Yes

Local: Yes

Published: May 01 2017 12:00AM/ Updated: May 08 2017 12:07AM

Credit: Maksim Malyutin from Embedi

Vulnerable: Lenovo ThinkStation S30 ,…

Not Vulnerable: Intel Standard Manageability 9.5.61.3012,….

Example vulnerabilities: CVE-2017-6074

Description:

The dccp_rcv_state_process function in net/dccp/input.c in the Linux kernel through 4.9.11 mishandles DCCP_PKT_REQUEST packet data structures in the LISTEN state, which allows local users to obtain root privileges or cause a denial of service (double free) via an application that makes an IPV6_RECVPKTINFO setsockopt system call

References:

http://www.securityfocus.com/bid/96310/info

CONFIRM:https://github.com/torvalds/linux/commit/5edabca9d4cff7f1f2b68f0bac55ef99d9798ba4

Vulnerable:

Ubuntu Ubuntu Linux 12.04 - 16.04 LTS, Redhat Enterprise Linux 5-7, Linux kernel 2.6 - 4.4.30, Debian Linux 6.0 *, CentOS CentOS 5 -7,...

The EGI Software Vulnerability Group (SVG)

- https://wiki.egi.eu/wiki/SVG:SVG
- The purpose of the EGI Software Vulnerability Group is "To minimize the risk to the EGI infrastructure arising from software vulnerabilities“
- The EGI SVG runs a procedure for handling software vulnerabilities reported which are relevant to the EGI infrastructure. This includes vulnerabilities announced by major providers, as well as software which is developed by collaborating projects and organisations used in the EGI infrastructure.
- Advisories are issued by SVG as part of this process.

Example vulnerabilities: CVE-2016-10030/ EGI-SVG-CVE-2016-1003

Affected software and risk:

'HIGH' risk privilege escalation vulnerability affecting the Linuxkernel n_hdlc module

Description:

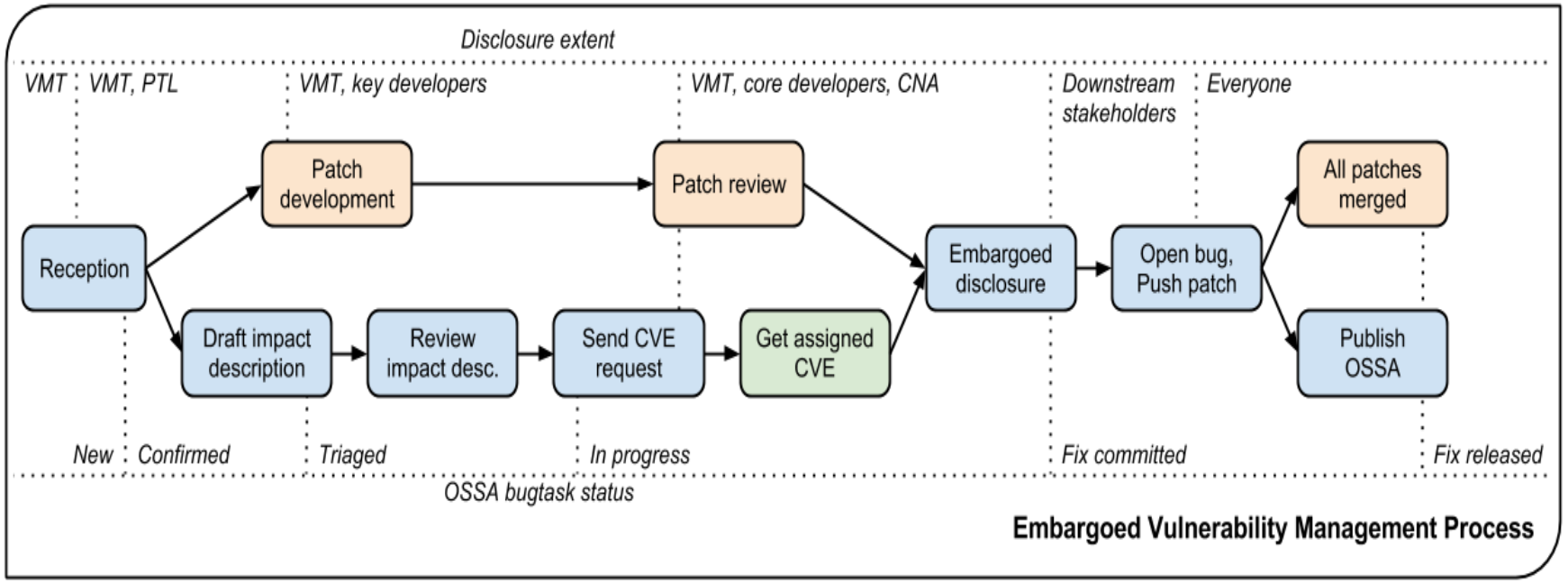
A local privilege escalation race condition in n_hdlc in linux kerneldriver has been found.
This vulnerability is present in all recent versions of the linuxkernel prior to the patched versions.
The most affected services are those that give shell access to unprivileged users:
- Worker Nodes
- shared User Interface hosts…

Actions required/recommended

Sites should apply vendor kernel updates as soon as possible, if updates are available.
If updates are not available, sites should consider taking mitigating action.

More information can be found at [R 1], [R 2], [R 3]

The dynamics of information about vulnerabilities



https://security.openstack.org/vmt-process.html

heartbleed.com

The Heartbleed Bug

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

Security patching vs. service availability

	Louhi	Murska
Security	6.82%	12.98 %
Hardware	47.30%	61.35 %
Software	18.11%	11.48%
IT infra (NFS & al.)	7.75%	0.14 %
Data Center infra	0.35 %	14.01 %
Other	19.67%	0.04%
TOTAL	**100.00 %**	**100.00 %**

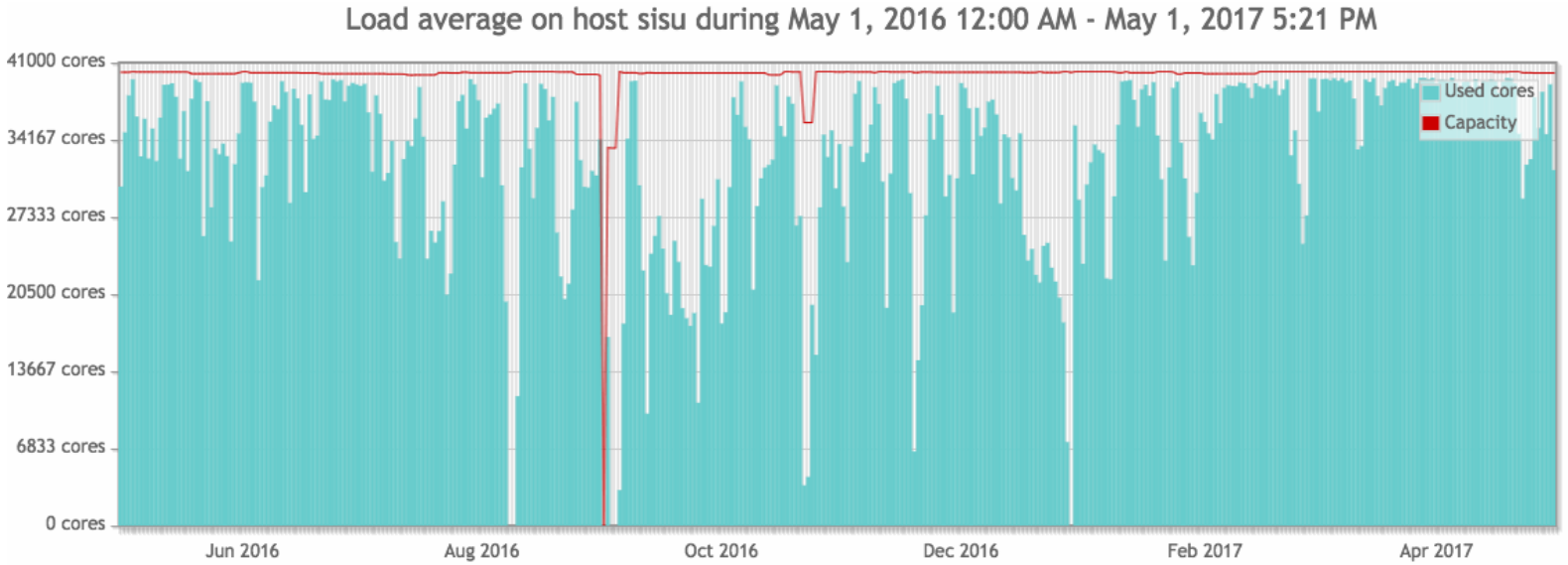
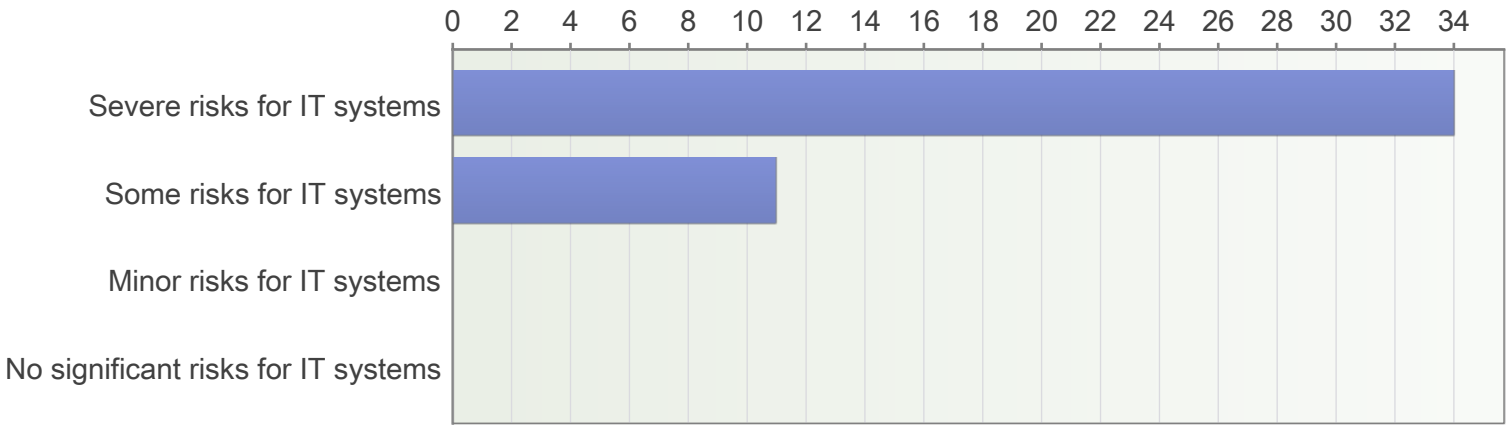


Table 6: System availability 2008-2011 (1.1.-30.4.2011).

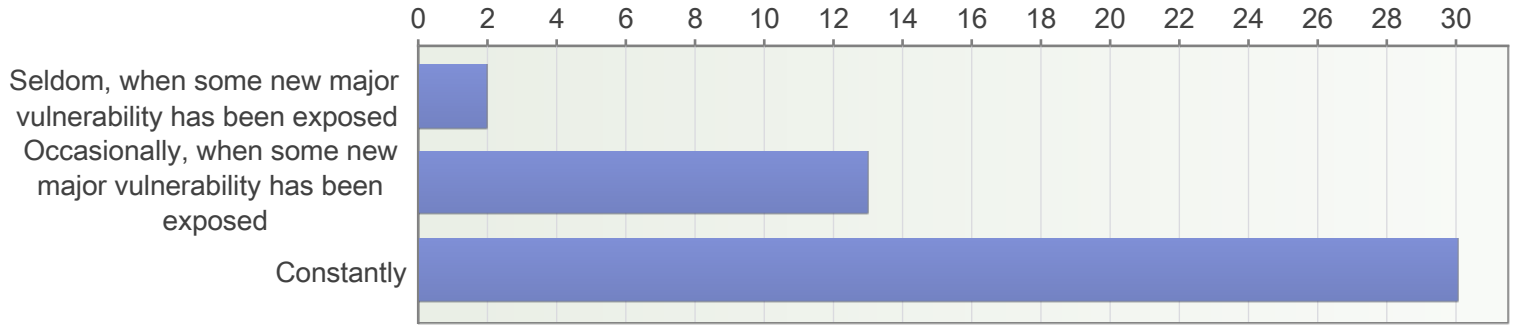
Metrics and Best Practices for Host-based Access Control to Ensure System Integrity and Availability. Urpo Kaila, Marco Passerini and Joni Virtanen. CUG 2011

- Results from a quick survey to system administrators

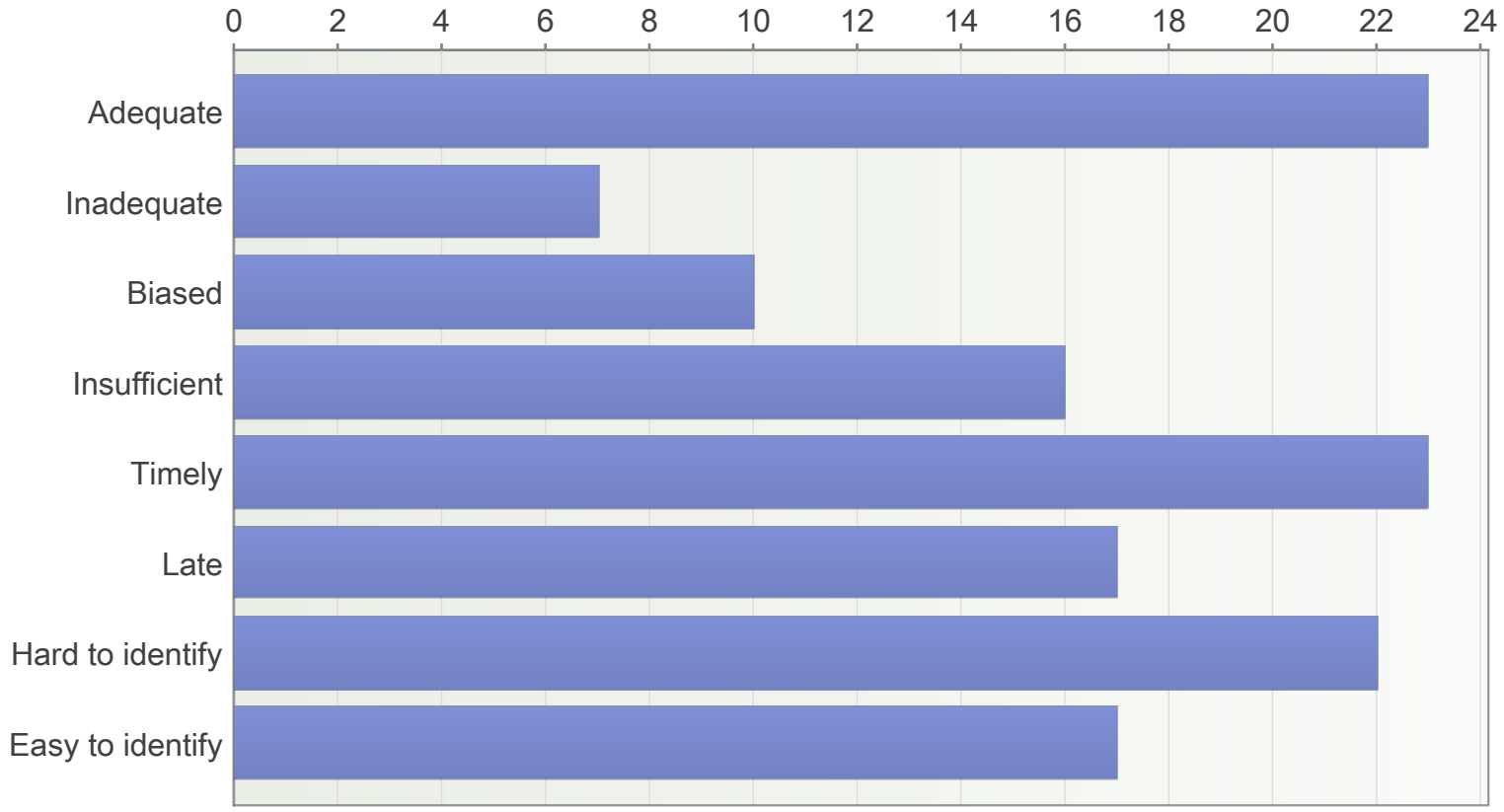
1. Software vulnerabilities can cause typically



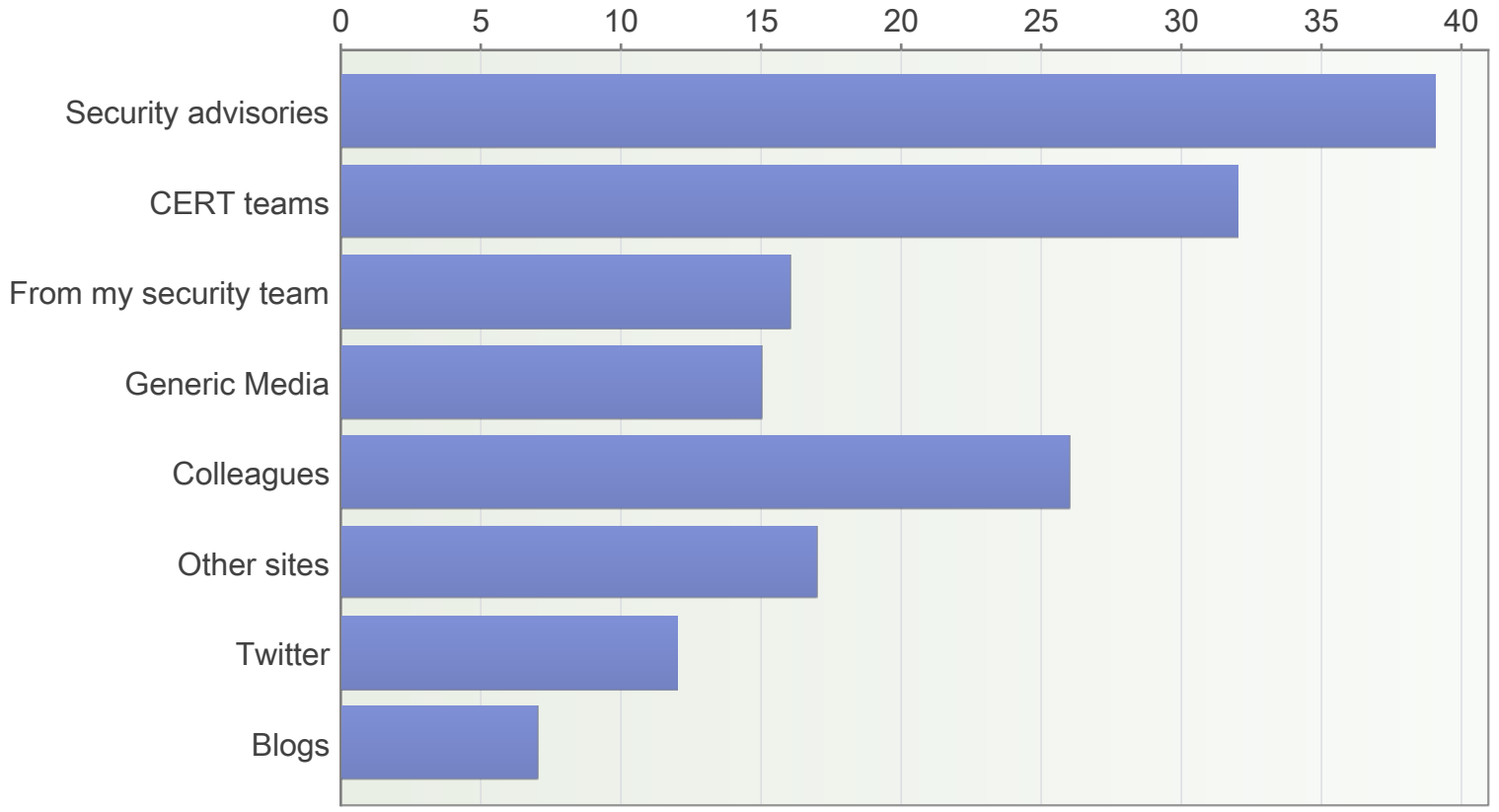
2. The risk caused by software vulnerabilities appears



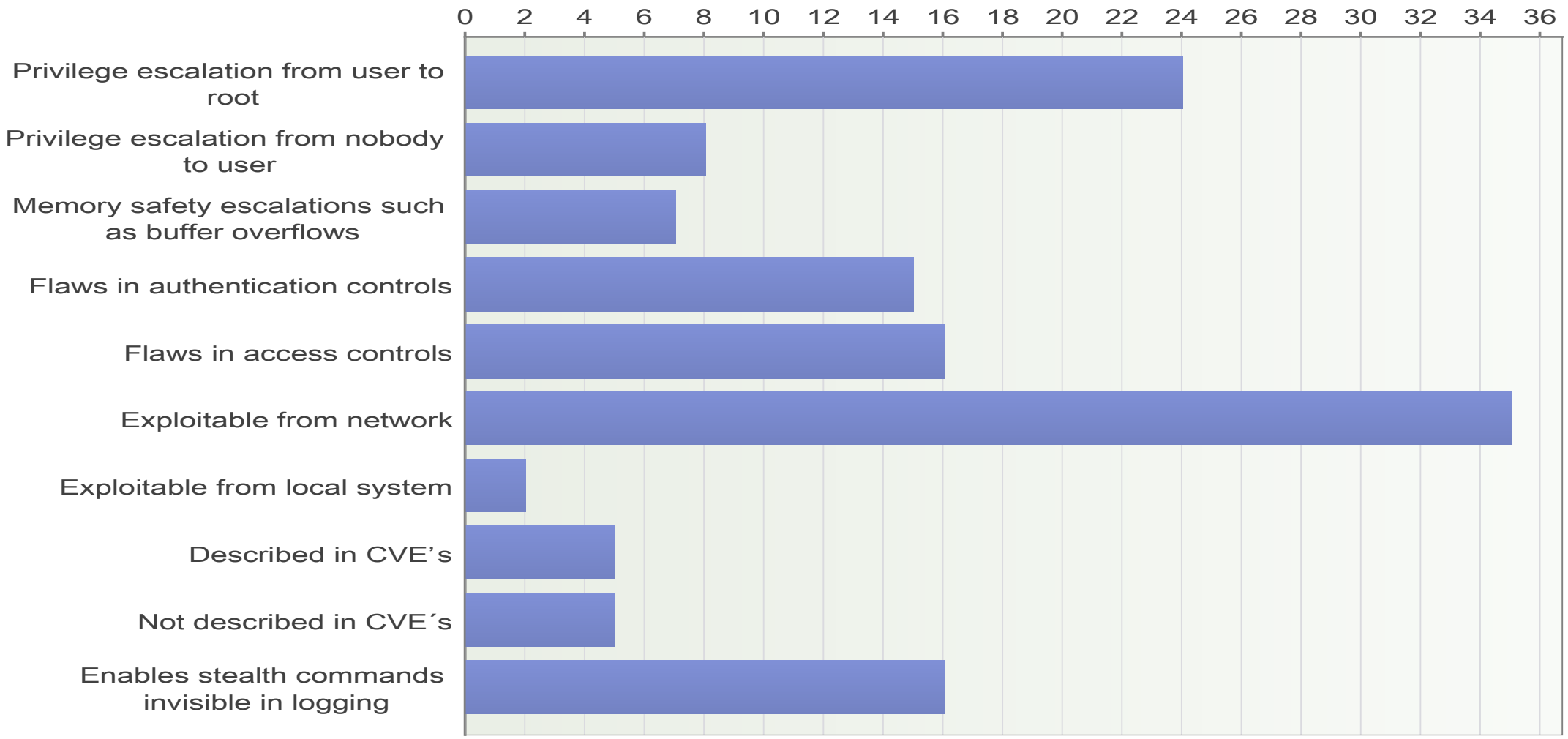
3. The information I receive about software vulnerabilities is (choose three)



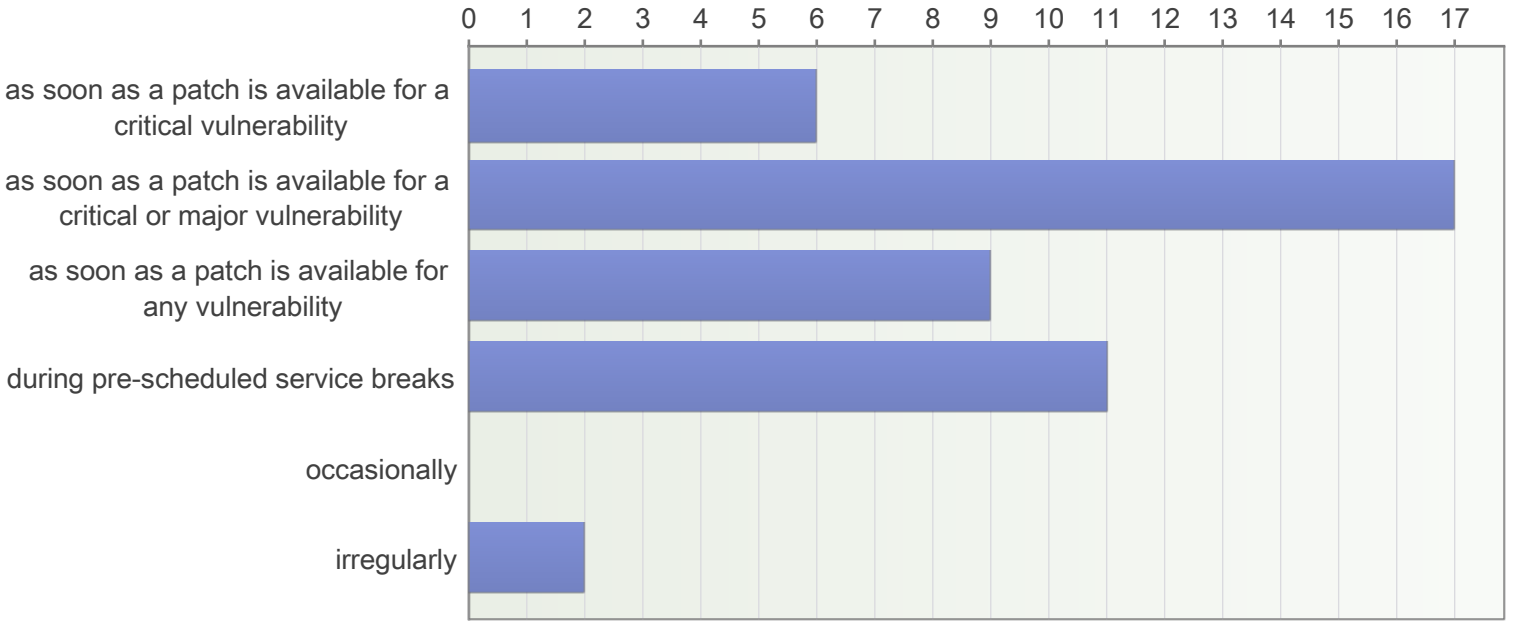
4. I receive information about vulnerabilities mainly from



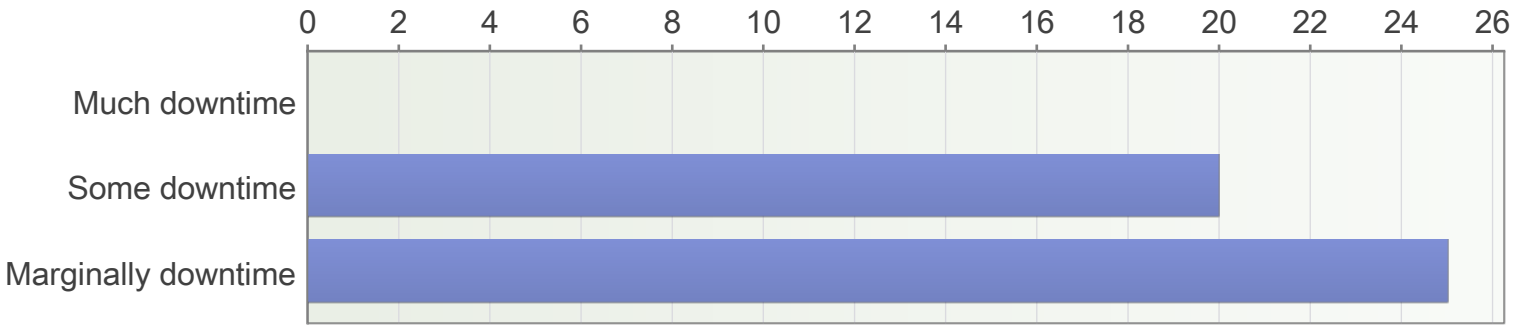
5. The most dangerous software vulnerabilities are (choose three)



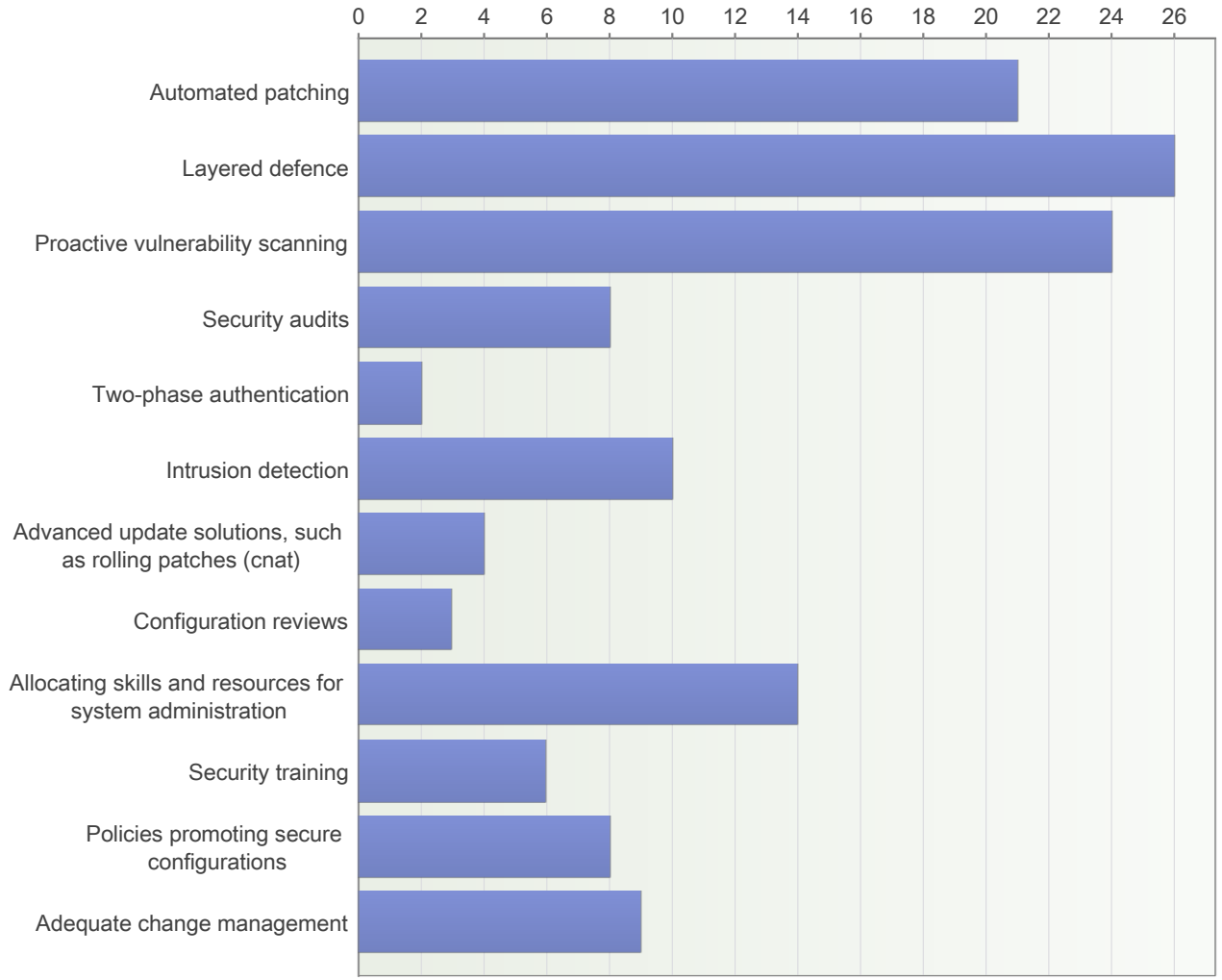
7. Software vulnerabilities and applying security patches typically causes



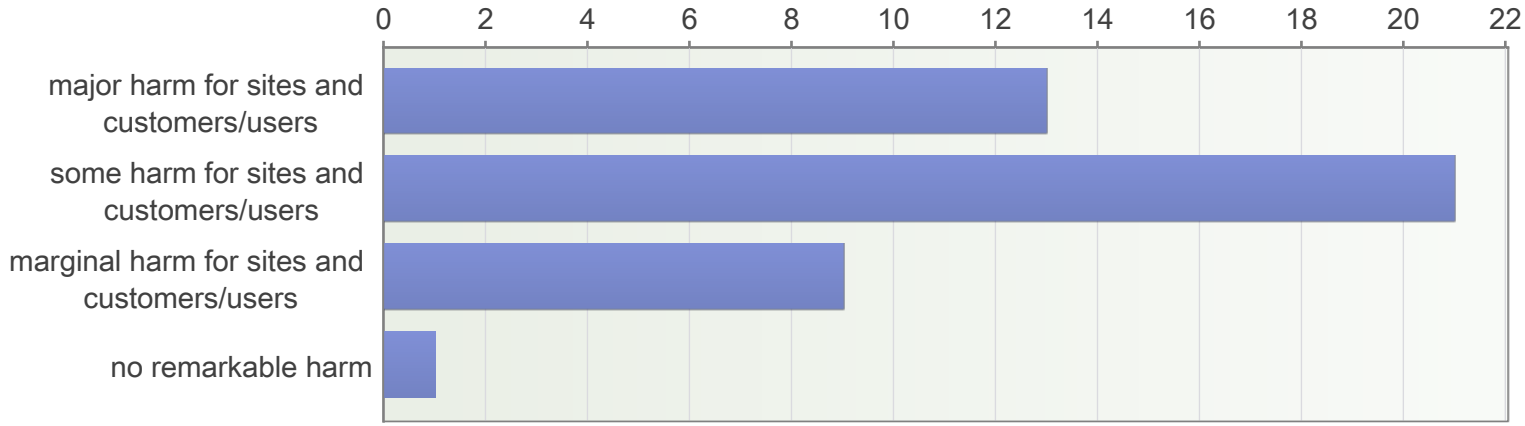
6. Patching of a security vulnerabilities are typically done



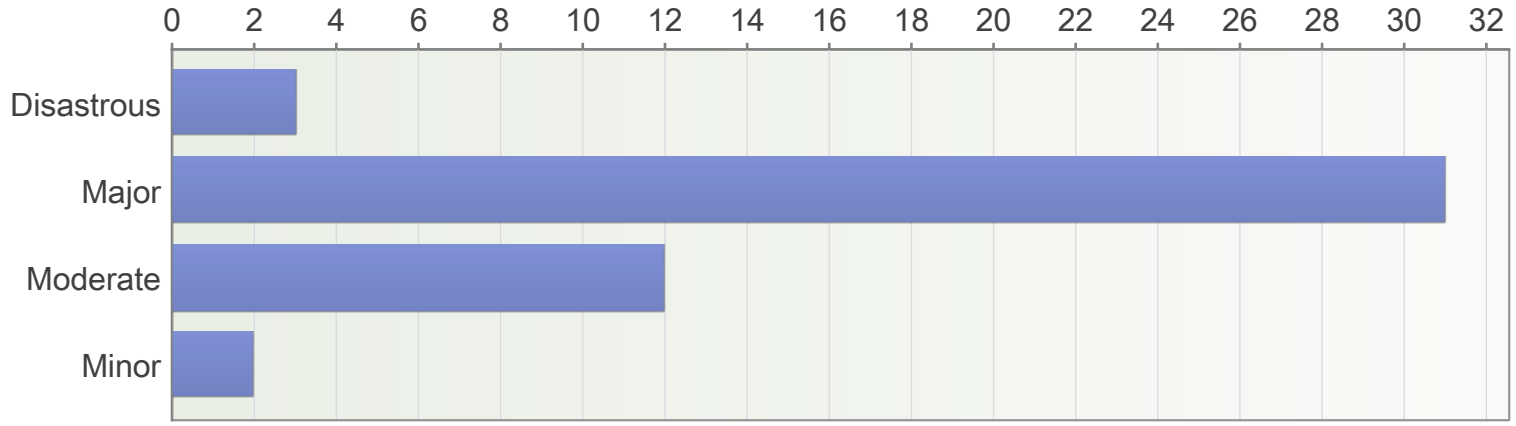
9. The best way to mitigate risks caused by software vulnerabilities are (choose three)



8. Intrusions and system compromises caused by software vulnerabilities will on yearly base result in



10. If a system is compromised by exploiting a software vulnerability, the impact is typically



Limitations of the query and comments

- Biased and limited population, CUG admins have unfortunately not been yet reached
- Some ambiguity in the question/ context can be understood in various ways
- No basic statistical metrics yet
- Comments from the respondents:
  - The survey is not just about software vulnerability, but about security vulnerability in general. To me software vulnerabilities are in general caused by bugs within the software, which would limit the scope. Security vulnerabilities can be caused by many different reasons, system, kernel and software bugs, system and network issues, misconfigurations,…
  - In a large organization you have many types of devices… as well as the mixed nature of configuration it is a complex problem to keep up to date with vulnerabilities….
  - There is also a large number of people of varying skill level
  - The balance is very much in the favor of the attacker,..
  - Even with vulnerabilities in software there should be several other layers to mitigate that.
- -> The query can just indicate views of a limited population

Best Practices/ Comments based on experience and on interviews*

- Subscribe to platform specific vulnerability advisories, national Cyber security agencies will seldom notify about vulnerabilities for specific systems
- Always react to all vulnerabilities, but take special care of those for which an exploit code is available (although it doesn't't currently work, it can be improved fast)
- Beware of vulnerabilities hyped in common media , there will soon be exploit codes for these
- Odd processes, strange network traffic, suspicious software executions can be a sign of an system intrusion – although these are also quite normal phenomena in research computing
- You don't need to be root to install application software, limited privileges are sufficient – but the administrators do need root access to be able to do all admin tasks

Special thanks to Mr. Esko Keränen

Best Practices/ Comments based on experience and on interviews

- Keep your system secure (basics):
  - always up-to-date patched
  - adequate authentication and access controls
  - shut down unnecessary services
  - enable strict firewall rules,
  - check that you don't have any dormant test or service accounts enabled
- Use standard unix procedures to protect users from each others
  - By default access to other users files must be restricted to administrators only
- Ensure that your system administrators have received adequate training and gained enough experience for secure administration practices – you must know what you do
- Apply strict access controls also - and specially – for administrators
- It is difficult to protect your system against bad system administration

Summary and Conclusions 1/2

- Protection against risks based on software vulnerabilities at computing centres for research related is a complex and dynamic tasks
  - Advance risks can emerge on short notice
  - It can be difficult to obtain timely and adequate information about the vulnerabilities which could endanger the system
  - It can also be difficult to identify anomalies indicating exploits in a dynamic and complex environment
- Implementing best security pays off
  - Adequate training and skills development for system administrators
  - Layered defence
  - Automated advanced patching – although patching cause some downtime
  - Vulnerability scanning
  - Resilient host based and network based access controls
  - A system intrusion would result in a major negative impact for the site

Summary and Conclusions 2/2

- IT Systems and services should be secure by design!
  - Explicit risk assessment
  - Specifying security requirements
  - Designing security architecture
  - Compliance/Vulnerability testing – and security certification
  - Operational security included in service management
    - Change management
    - Incident management
    - Vulnerability management
    - Monitoring
    - To be implemented on team and on individual level
    - To be supported by adequate tools and procedures

- Better technologies needed to cope with kernel level vulnerabilities
  - Polite patch-and-reboot with fail active operational with solutions like kpatch
  - Redundancy optimization
  - Job hibernation and restore from last known good "dbcc"

Acknowledgments and thanks

- Special thanks for comment and advise to
  - Mr. Esko Keränen, CSC iconic and retired Cray Administrator
  - Dr Linda Cornwall, STFC Rutherford Appleton Laboratory/ EGI Software Vulnerability Group

- Many thanks for comments and advise to colleagues and friends at
  - WISE - https://wise-community.org
  - GÉANT SIG-ISM - https://www.geant.org/Innovation/SIG_TF/Pages/SIG-ISM.aspx
  - CUG - https://cug.org/
  - Funet Security Team
  - CSC - https://www.csc.fi
  - EUDAT – https://www.eudat.eu
  - EGI - https://www.egi.eu/

- Thank you! Questions and comments welcome – urpo.kaila@csc.fi