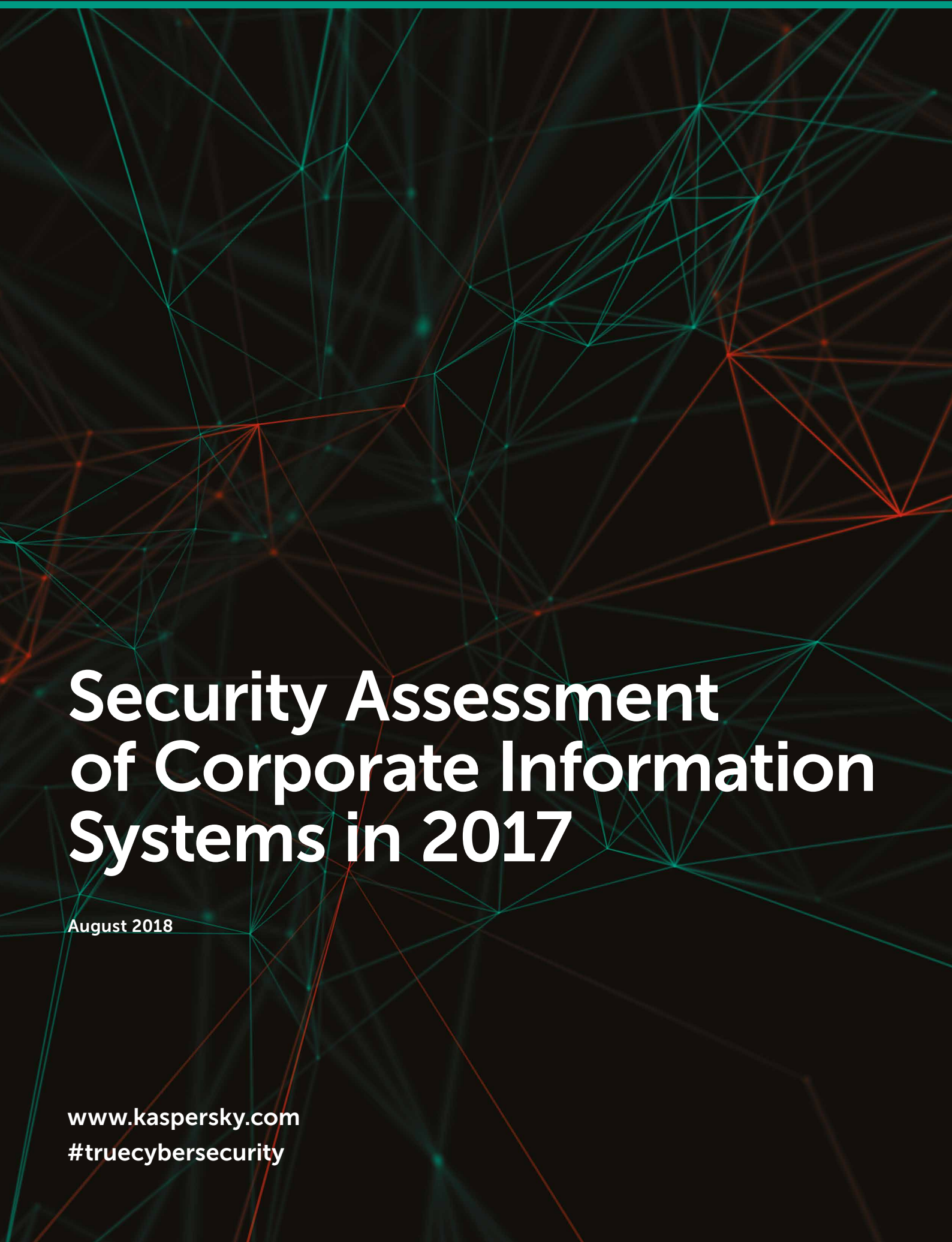
KASPERSKY lab



Security Assessment of Corporate Information Systems in 2017

August 2018

www.kaspersky.com
#truecybersecurity

Contents

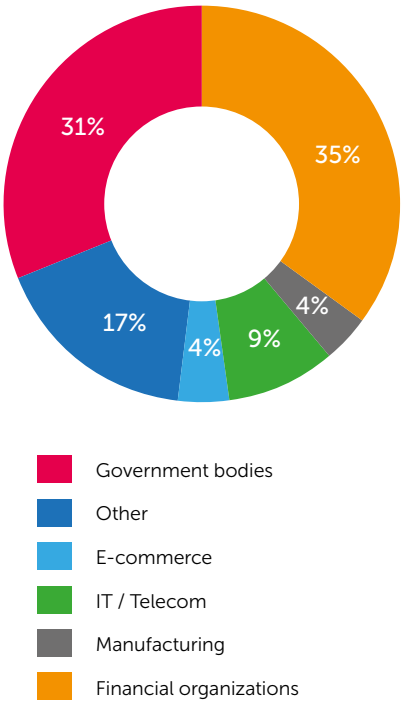
Introduction

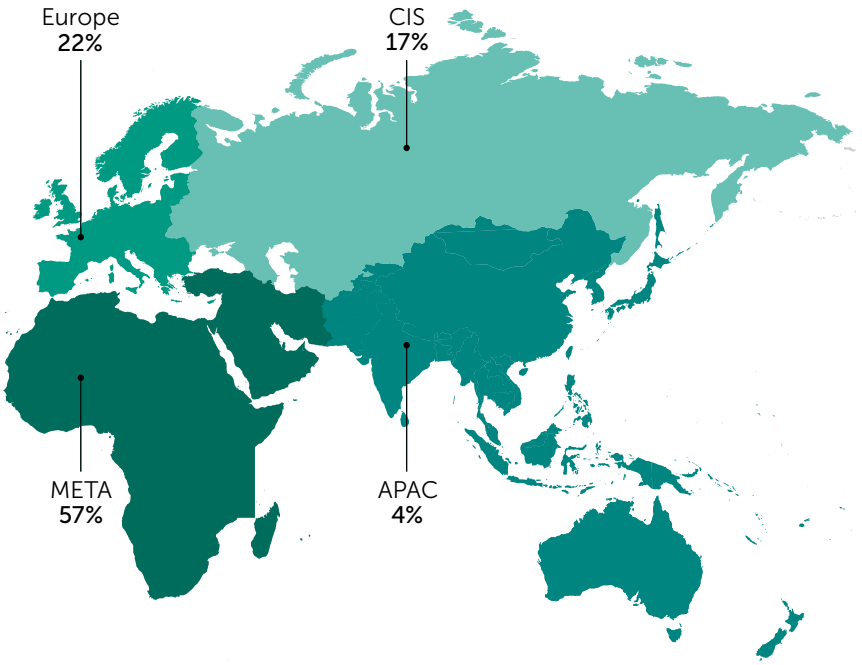
Each year, Kaspersky Lab's Security Services department carries out dozens of cybersecurity assessment projects for companies worldwide. In this publication we present a general summary and statistics for the cybersecurity assessments of corporate information systems Kaspersky Lab has conducted throughout 2017.

The primary goal of this publication is to offer information support to IT security specialists in the area of vulnerabilities and attack vectors against modern corporate information systems.

We have analyzed several dozens of projects for companies from various sectors, including government bodies, financial organizations, telecommunication and IT companies, as well as manufacturing and energy companies. The charts below demonstrate the distribution of the analyzed companies by industry and by region.

The distribution of the analyzed companies by industry and by region





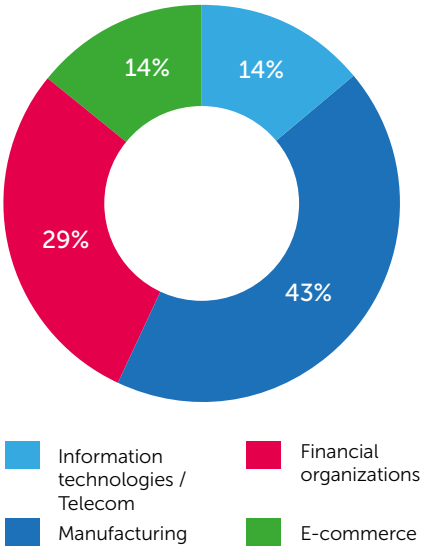
The summary and statistics on detected vulnerabilities are provided separately for each type of service provided:

- **External penetration testing** is an assessment of an organization's cybersecurity posture when challenged by an external intruder from the Internet who only has access to publicly available information.
- **Internal penetration testing** is an assessment of an organization's cybersecurity posture when challenged by a threat actor who is located inside the client's corporate network, has physical access to the analyzed objects only and has no privileges on the internal network.
- **Web application security assessment** is the search for vulnerabilities and security flaws resulting from mistakes made during the design, development or operation of a web application.

This publication includes statistics on the most common vulnerabilities and security flaws that Kaspersky Lab's experts have detected and that can potentially be used by threat actors for unauthorized penetration into company infrastructures.

Assessment of protection against external intruders

Analyzed companies by economic sector



Organizations were assessed for security levels on the following scale:

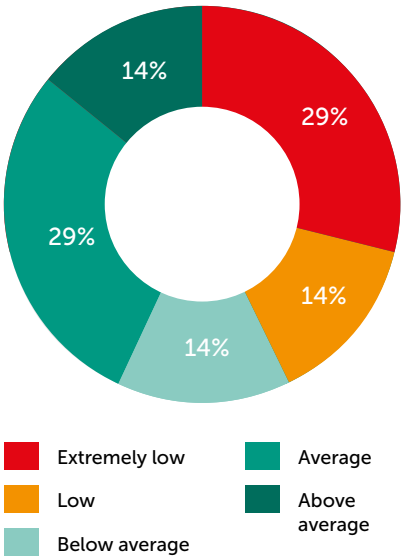
- **Extremely low**
- **Low**
- **Below medium**
- **Medium**
- **Above medium**
- **High**

The overall security levels were assessed using Kaspersky Lab's own methodology which takes into account the level of access gained during testing, the priorities of the information resources, how difficult it was to gain access and the time it took.

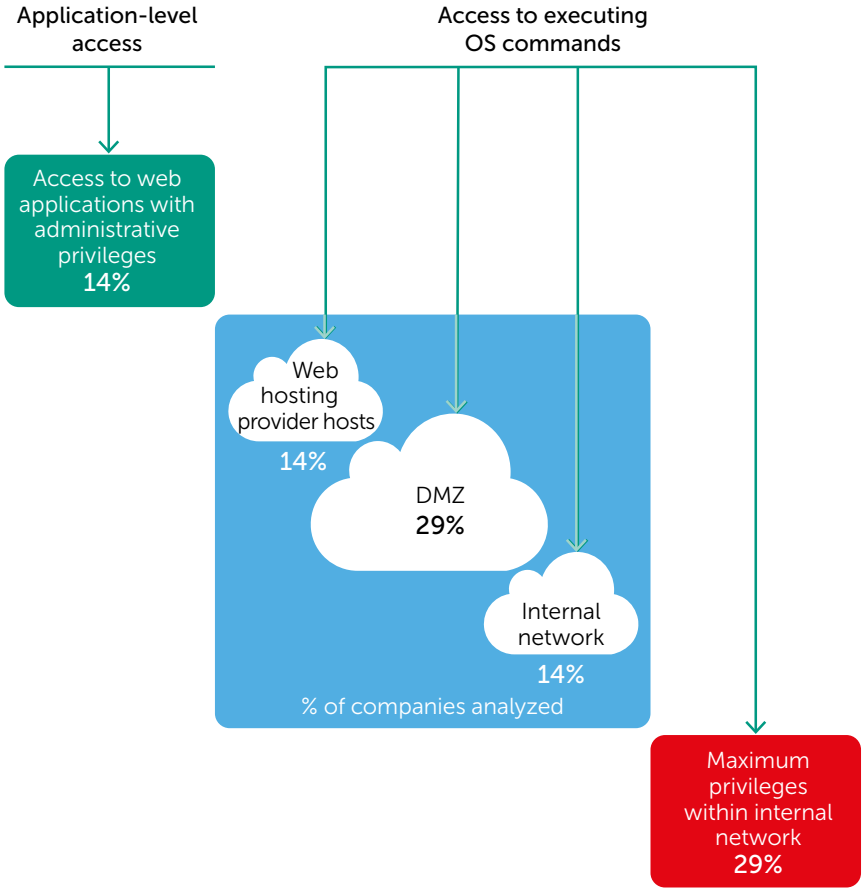
An extremely low level of protection corresponds to those cases where we were able to penetrate the network perimeter and gain access to the critical resources of the internal network (i.e. gain maximum privileges in the internal network, gain complete control over key business systems and access critical information). Moreover, gaining such access does not require special skills or a lot of time.

A high level of protection corresponds to those cases where only insignificant vulnerabilities were identified at the client's network perimeter, the exploitation of which does not carry risks for the company.

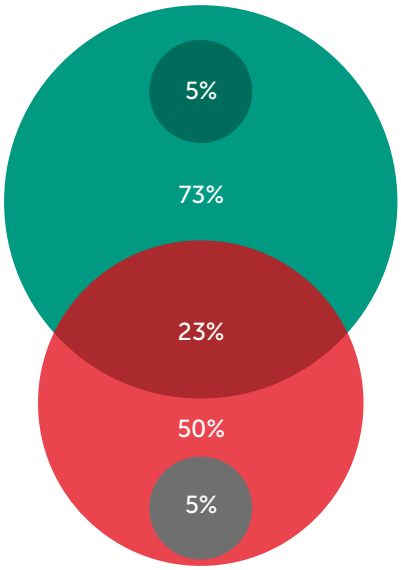
Company security levels



Distribution of analyzed companies according to access level gained during testing



Attack vectors used to penetrate network perimeter



- Exploitation of known vulnerabilities in web components (web servers, libraries)
- Attacks via web application vulnerabilities
- Attacks via management interfaces and via exploitation of web applications
- Attacks via management interfaces
- Exploitation of known vulnerabilities in software management interfaces

Most of the attack vectors that were successfully implemented were possible because of inadequate network filtering, publicly available network access to management interfaces, weak account passwords and vulnerabilities in web applications.

Although 86% of all analyzed companies were found to use obsolete, vulnerable software, only 10% of attack vectors used to penetrate the network perimeter (28% of analyzed companies) exploited vulnerabilities related to the absence of the latest software updates. This is due to the fact that exploitation of such vulnerabilities may cause denial of service. These limitations to attack demonstrations are caused primarily by the peculiarities of penetration testing services – it is a priority to keep the client's resources operational. Real cybercriminals, however, may ignore such considerations when launching attacks.

> Recommendation:
>
> As well as update management, pay more attention to configuring network filtering rules, password protection and eliminating vulnerabilities in web applications.

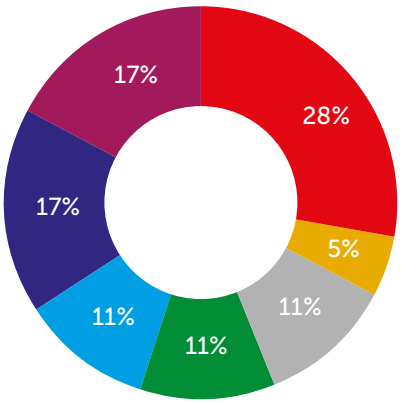
Attacks via vulnerabilities in web applications

The results of penetration tests performed in 2017 show unambiguously that insufficient attention is paid to the security of web applications. In 73% of the implemented attack vectors, web app vulnerabilities were used to gain access to network perimeter hosts.

Arbitrary file upload was the most widespread web app vulnerability used to penetrate the network perimeter during penetration tests. It was used to upload a command line interpreter and gain access to the operating system. Vulnerabilities of the types 'SQL injection', 'Arbitrary file reading', 'XML external entity' were used predominantly to obtain sensitive information, such as passwords or their hashes. Account passwords were used to develop attacks via publicly available management interfaces.

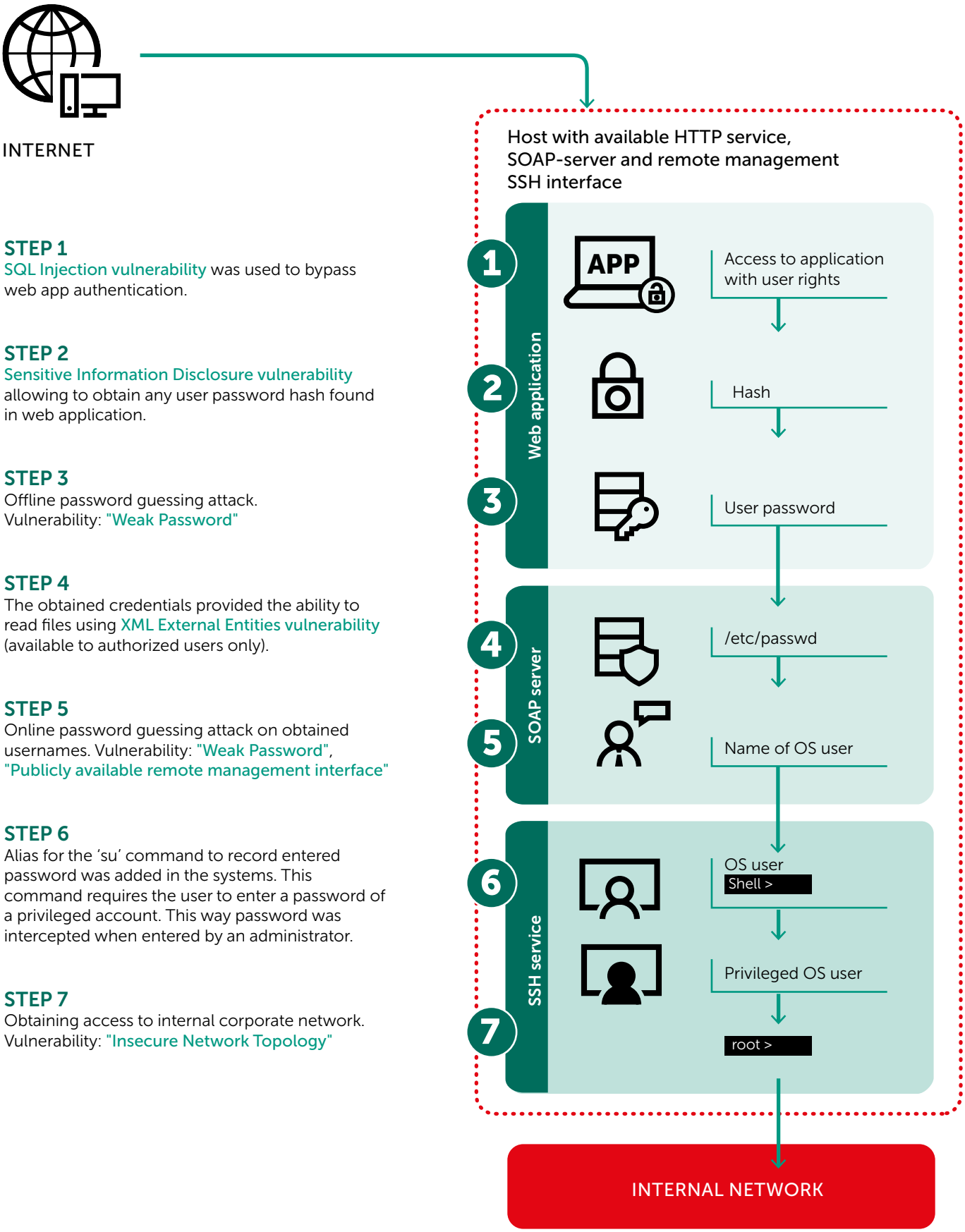
> Recommendation:
>
> A security assessment should be performed on a regular basis for all publicly available web applications. A vulnerability management process should be implemented; applications must be checked after changes are introduced into application code or web server configuration; third-party components and libraries must be updated in a timely manner.

Web application vulnerabilities used to penetrate network perimeter



- Arbitrary File Upload
- XML External Entity
- Other
- Arbitrary File Creation/Modification
- Arbitrary File Reading
- SQL Injection
- Code Injection

Example of how access is gained to internal network via web app vulnerabilities and publicly available management interface



STEP 1
SQL Injection vulnerability was used to bypass web app authentication.

STEP 2
Sensitive Information Disclosure vulnerability allowing to obtain any user password hash found in web application.

STEP 3
Offline password guessing attack.
Vulnerability: "Weak Password"

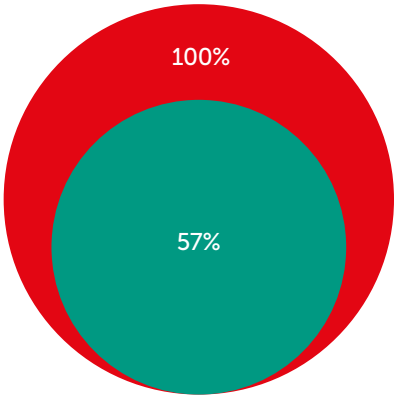
STEP 4
The obtained credentials provided the ability to read files using XML External Entities vulnerability (available to authorized users only).

STEP 5
Online password guessing attack on obtained usernames. Vulnerability: "Weak Password", "Publicly available remote management interface"

STEP 6
Alias for the 'su' command to record entered password was added in the systems. This command requires the user to enter a password of a privileged account. This way password was intercepted when entered by an administrator.

STEP 7
Obtaining access to internal corporate network.
Vulnerability: "Insecure Network Topology"

Attacks via management interfaces



Companies with publicly available management interfaces

Companies whose resources were accessed via management interfaces

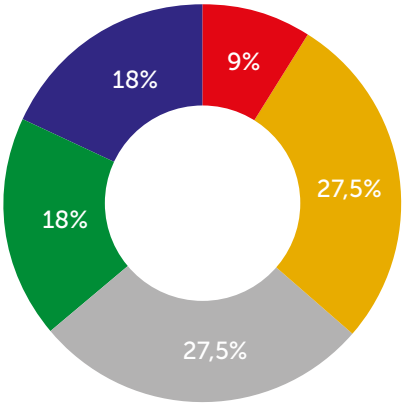
Although 'Unrestricted network access to management interfaces' is not a vulnerability but only a configuration flaw, it was used in half of all attack vectors implemented in 2017 during external penetration tests. Access to the information resources of 57% of analyzed companies was gained via management interfaces.

Access via management interfaces was most often gained using passwords which were obtained by:

- **Exploiting other vulnerabilities of the attacked host (27.5%).** For example, exploitation of an 'Arbitrary file Reading' vulnerability in the web application allowed the attacker to obtain a clear-text password from the web application's configuration file.
- **Using default credentials for web applications, CMS systems, network devices, etc. (27.5%).** The attacker can find the account credentials required for access in the appropriate documentation.
- **Launching an online password guessing attack (18%).** When there is no protection in place against such attacks and/or tools to detect them, the attacker's chances of guessing the password improve dramatically.
- **Credentials obtained from another compromised host (18%).** Using the same passwords for multiple systems expand the potential attack surface.

Exploiting known vulnerabilities in obsolete software was the least common scenario when access was gained via management interfaces.

Gaining access via management interfaces



Exploitation of known vulnerabilities in the exposed service's software (obsolete software)

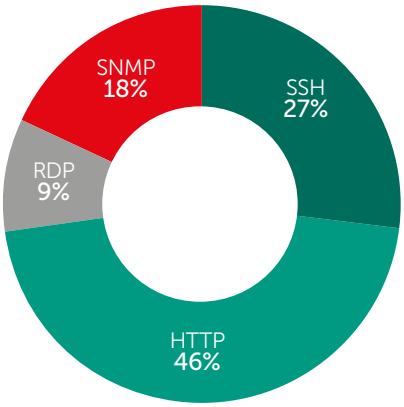
Credentials obtained via another vulnerability

Default credentials

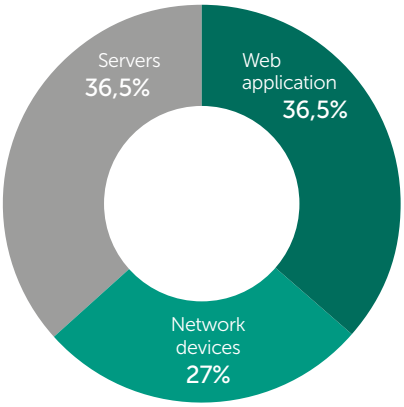
Password received from another host

Brute force attack on user account

Management interfaces via which access was gained



Management interfaces



Recommendation:

Check all systems on a regular basis, including web applications, content management systems and network devices, to see if any default credentials are being used. Set strong passwords for administrative accounts. Use different accounts for different systems. Update software to the latest versions.

Most often, companies forget to block network access to the remote management web interfaces and to the SSH service. Most management web interfaces are administration control panels of web applications or a CMS. Access to an application's administration control panel often makes it possible not only to gain complete control over the web application but also gain access to the operating system. After gaining access to a web application's administration control panel, access to execute operating system commands can be gained using the arbitrary file upload capability or by editing the web application's pages. In some cases, a command line interpreter is a built-in feature in the web application's administration control panel.

Recommendation:

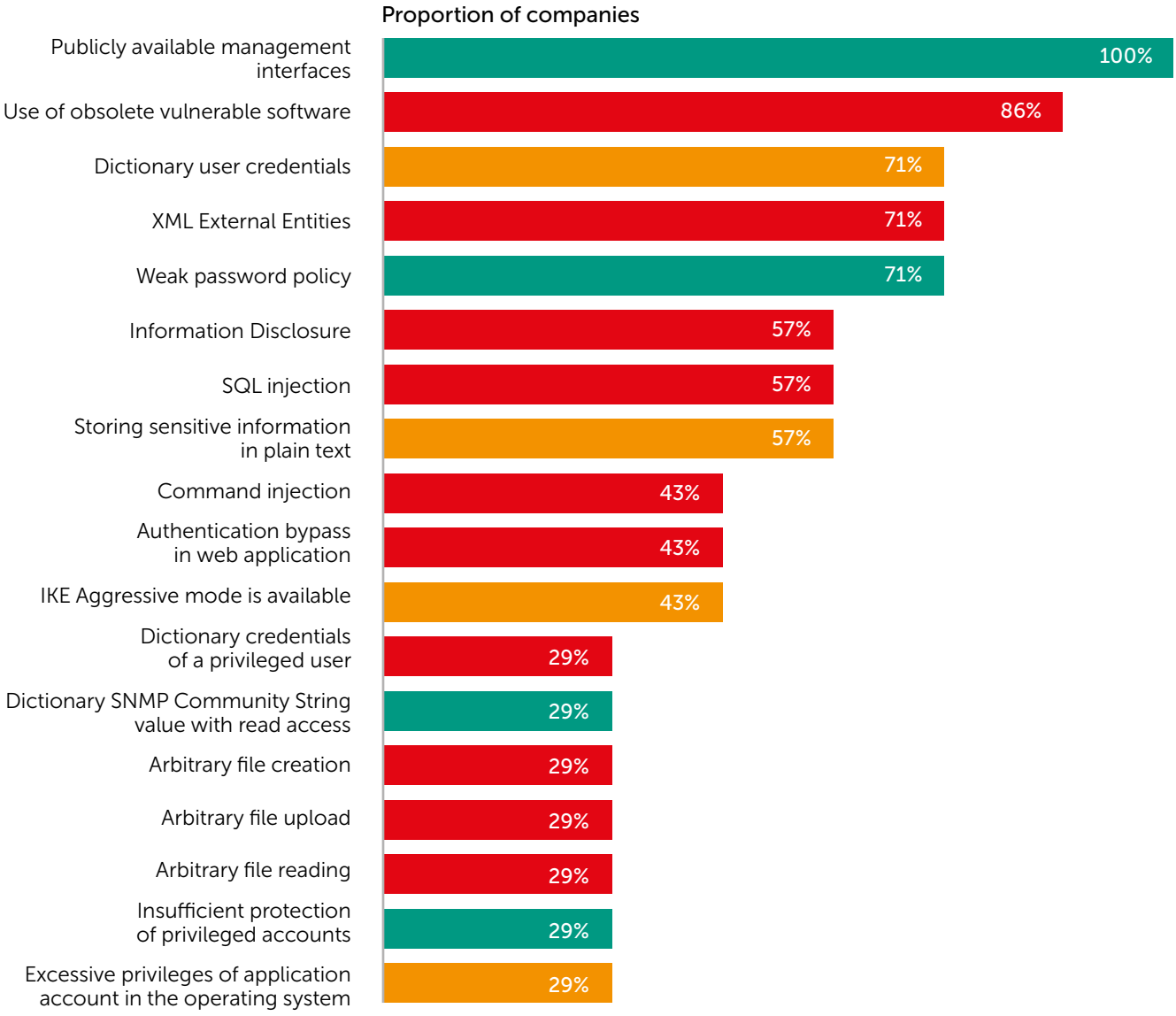
Restrict network access to all management interfaces, including web interfaces. Access must only be allowed from a restricted number of IP addresses. Use VPN for remote access.

Example of an attack via management interfaces

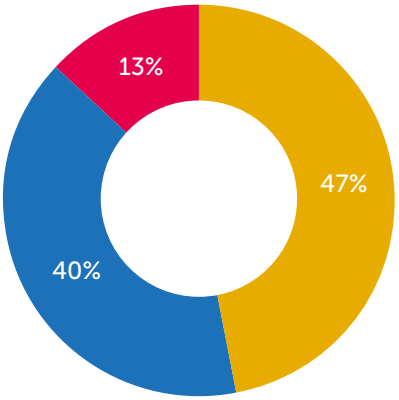
Step 1	An SNMP service which has default community string with read-only access is detected
Step 2	It has been detected via SNMP protocol that an obsolete, vulnerable version of Cisco IOS is being used. Vulnerability: cisco-sa-20170629-snmp. (https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20170629-snmp). This vulnerability allows the attacker with a read-only SNMP community string to gain fully privileged access to a device. Using the general information about the vulnerability that was published by Cisco, Artem Kondratenko, senior pentesting specialist at Kaspersky Lab, has developed an exploit (https://github.com/artkond/cisco-snmp-rce) with which the attack can be demonstrated in practice.
Step 3	Exploitation of a vulnerability in ADSL-LINE-MIB and the gaining of fully privileged access to a router allowed us to gain access to the client's internal network resources. For the technical details of the exploit for this vulnerability, please visit: https://kas.pr/3whh

Statistics on the most common vulnerabilities and security flaws

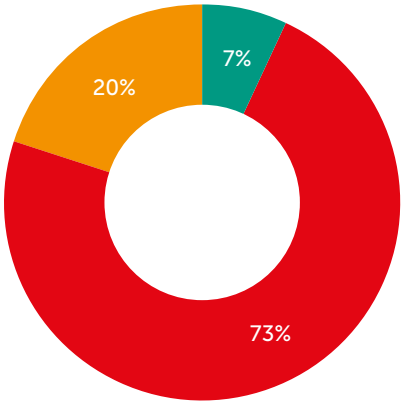
The most common vulnerabilities and security flaws



Assessment of protection against internal intruders



Financial organizations
Government bodies
Manufacturing



Extremely low
Low
Above medium

Organizations were assessed for security levels on the following scale:

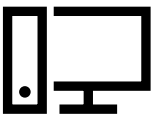
- **Extremely low**
- **Low**
- **Below medium**
- **Medium**
- **Above medium**
- **High**

The overall security levels were assessed using Kaspersky Lab's own methodology which takes into account the level of access gained, the priority of the information resources, how difficult it was to gain access and the time it took. An extremely low level of protection corresponds to those cases where we were able to obtain total control over the customer's internal network (i.e. gain maximum privileges in the internal network, gain complete control over key business systems and access critical information). Moreover, gaining such access does not require special skills or a lot of time.

A high level of protection corresponds to those cases where penetration testing identifies only insignificant vulnerabilities in the internal network resources, the exploitation of which does not pose serious risks to information security.

Maximum privileges in the Active Directory domain (i.e. Domain Administrator or Enterprise Administrator privileges) were gained in 86% of all projects where domain infrastructure was present. In 64% of companies, more than one attack vector was identified via which maximum privileges could be gained. In each project, an average of 2-3 attack vectors were identified via which maximum privileges could be gained. Only those attack vectors which were demonstrated in practice during the provision of internal penetration testing service were counted. For most projects lots of other potential attack vectors were also identified using specialized tools such as bloodhound.

Privileges gained in 86% of analyzed companies



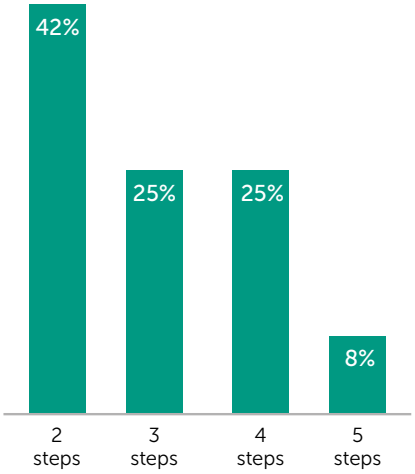
AUDITOR

On average, 2-3 vectors identified in each company

On average, 3 steps in a vector

DOMAIN ADMINISTRATOR/ ENTERPRISE ADMINISTRATOR

Minimum number of steps to gain domain administrator privileges



The attack vectors that were demonstrated in practice varied in terms of their complexity and the number of steps (from 2 to 6) it took to implement them. On average, it took 3 steps to gain Domain Administrator privileges in each company.

Examples of the easiest attack vectors with which domain administrator privileges were gained:

- Launching a combination of NBNS Spoofing and NTLM Relay attacks allows the threat actor to intercept the administrator's NetNTLM hash and use it to authenticate at the domain controller.
- Exploitation of the vulnerability CVE-2011-0923 in HP Data Protector and subsequent extraction of the domain administrator's password from the memory of lsass.exe process.

The diagram below depicts an example of a more complex attack vector to gain domain administrator privileges by exploiting the following vulnerabilities:

- Usage of obsolete versions of network device firmware that contain known vulnerabilities.
- Use of weak passwords.
- Password reuse across multiple systems and users.
- Use of the NBNS protocol.
- Excessive privileges of an account with SPN.

Example of obtaining Domain administrator privileges

STEP 1
Exploitation of vulnerability in the web service of D-Link network storage. Vulnerability allows executing of arbitrary code with super-user privileges. Creating SSH tunnel to access management network (direct access is restricted by the firewall rules). Vulnerability: 'Obsolete software (D-link)'.

STEP 2
Detected Cisco switch, which had an available SNMP service and the default community string "Public". Cisco IOS version was identified via SNMP protocol. Vulnerability: 'Default SNMP community string'.

STEP 3
Information about the Cisco IOS version was used to identify vulnerabilities. Exploitation of vulnerability CVE-2017-3881. Access to command interpreter with maximum privileges was obtained. Vulnerability: 'Obsolete Software (Cisco)'.

STEP 4
Extracting the hash value of the local user's password.

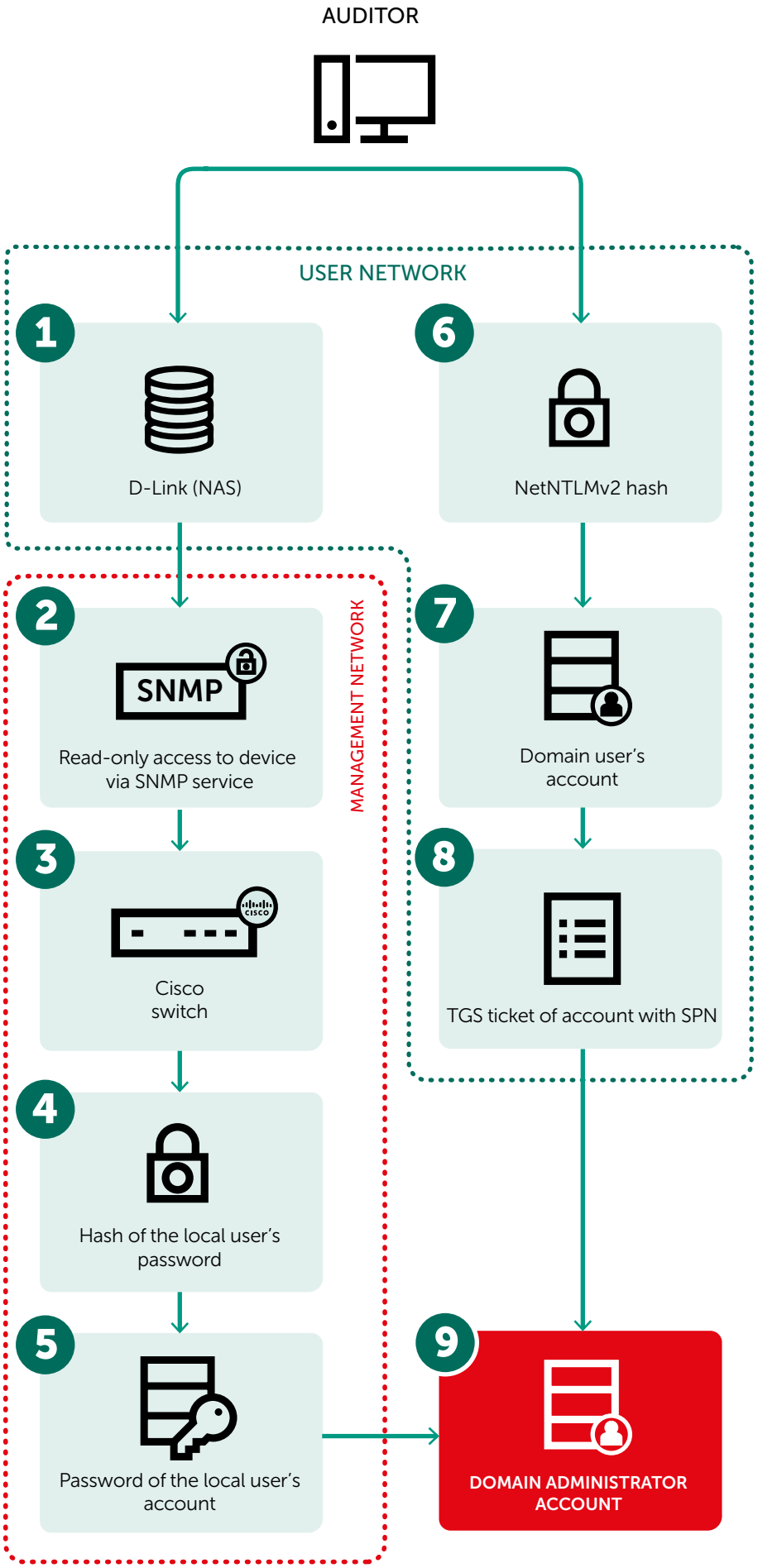
STEP 5
Offline password guessing attack. Vulnerability: 'Weak privileged user's password'.

STEP 6
Conducting an NBNS Spoofing attack. Intercepting NetNTLMv2 hash. Vulnerability: 'Use of NBNS protocol'.

STEP 7
Offline password guessing attack on NetNTLMv2 hash. Vulnerability: 'Weak user password'.

STEP 8
Domain user account was used to perform Kerberoasting attack. TGS ticket of the account with SPN obtained.

STEP 9
The password of the local user account obtained from the Cisco switch was the same as password of the account with SPN. Vulnerability: 'Password reuse', 'Excessive account privileges'.



About the vulnerability CVE-2017-3881 (Remote code execution in Cisco IOS)

A reference to this vulnerability was found in CIA documents **Vault 7: CIA**, which were published on WikiLeaks in March 2017. The vulnerability was codenamed ROCEM and there was virtually no description of its technical details. Later, identifiers CVE-2017-3881 and cisco-sa-20170317-cmp were assigned to it.

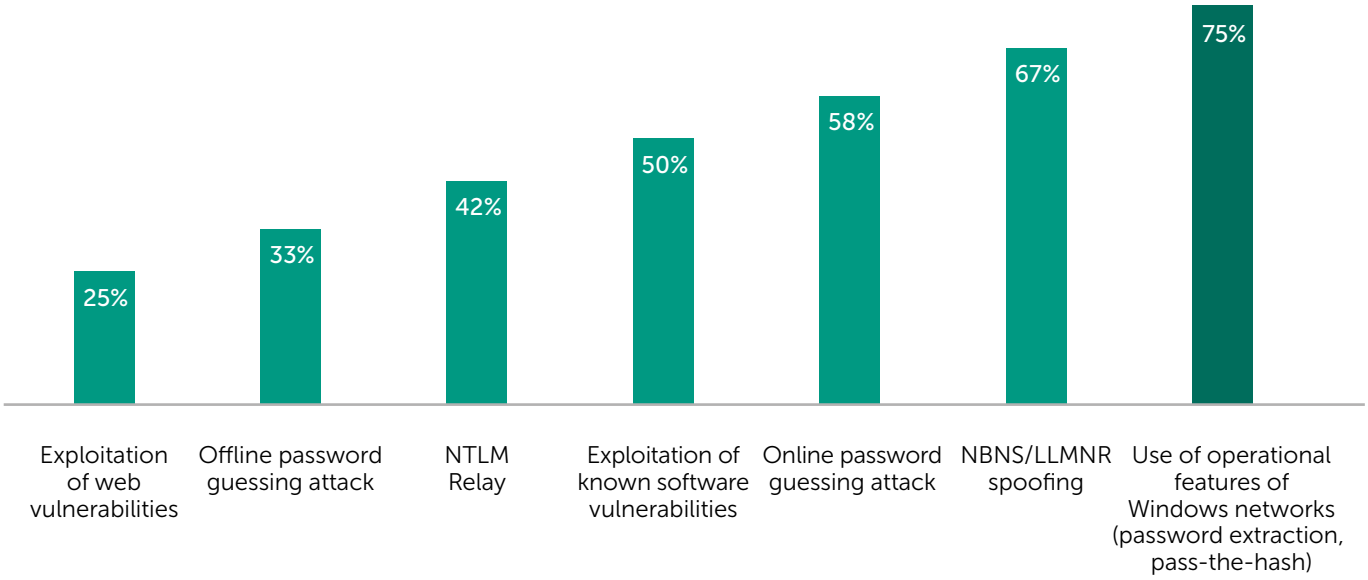
This vulnerability allows an unauthorized attacker to execute arbitrary code in Cisco IOS with maximum privileges via the Telnet protocol. In the CIA document, only some of the details were described that related to the testing process required to develop the exploit; however, the source code of the actual exploit was not provided. Nonetheless, Kaspersky Lab expert Artem Kondratenko was able to use the available information to conduct lab research and reproduce the exploit for this critical vulnerability.

For a detailed description of how the exploit for this vulnerability was developed, please visit https://kas.pr/fk8g, https://kas.pr/amv7.

Most commonly used attacks and techniques

Analysis of the attacks and techniques used to gain maximum privileges in the Active Directory domain yielded the following results:

Percentages of companies in which various attacks and techniques were used to gain maximum privileges in the Active Directory domain



NBNS/LLMNR Spoofing Attack

Successfully applied in
67%
of companies

87%
of companies
are vulnerable

We discovered that the protocols NBNS and LLMNR were used in 87% of the analyzed companies. In 67% of the companies where maximum privileges in the Active Directory Domain were gained, an NBNS/LLMNR Spoofing attack was successfully applied. With this type of attack user data, including users' NetNTLMv2 hashes, can be intercepted. This hash can be used to conduct a password guessing attack.

Security recommendations:

It is recommended to disable the NBNS and LLMNR protocols.

Recommendations for detection:

One possible solution is to use honeypots to broadcast NBNS/LLMNR requests with non-existing computer names over the network. If responses to these requests arrive, it indicates an attacker is present in the network. Examples: https://blog.netspi.com/identifying-rogue-nbns-spoofers/, https://github.com/Kevin-Robertson/Conveigh. If there is access to a copy of the entire network traffic, watch out for multiple LLMNR/NBNS responses sent from a single IP address in response to requests with different computer names.

NTLM Relay

Successfully applied in
42%
of companies

47%
of companies
are vulnerable

In half of all cases when an NBNS/LLMNR Spoofing attack was successful, the intercepted NetNTLMv2 hashes were used to conduct an NTLM Relay attack. If the NetNTLMv2 hash of the domain administrator account was intercepted during an NBNS/LLMNR Spoofing attack, then an NTLM Relay attack helps to rapidly gain maximum privileges in Active Directory.

An NTLM Relay attack (in combination with an NBNS/LLMNR Spoofing attack) was used to gain maximum privileges in the Active directory domain in 42% of the analyzed companies. 47% of the analyzed companies were found to have no protection against this type of attack.

Security recommendations:

The most effective method to protect against this attack is to block authentication via the NTLM protocol. A drawback to this approach is that it is difficult to implement.

Extended Protocol for Authentication (EPA) can be used to protect against an NTLM Relay attack.

Another protection mechanism may be to enable signing in the SMB protocol in group policies settings. Please note that this approach only protects against NTLM Relay attacks targeting the SMB protocol.

Recommendations for detection:

An indicator of this type of attack can be a network logon event (event 4624, Logon Type 3) in which the IP address in the field "Source Network Address" does not correspond to the source hostname "Workstation Name". At this stage, a table for mapping computer names to IP Addresses is required (integration with DNS can be used).

Alternatively, this sort of attack can be identified by monitoring network logons from non-typical IP addresses. For each network host, statistics should be collected on IP addresses from which system logon is performed most frequently. A network logon from a non-typical IP address may be indicative of a possible attack. A disadvantage to this approach is the large number of false positives.

Exploitation of known vulnerabilities in obsolete software

Successfully applied in
50%
of companies

Vulnerability MS17-010 identified in
75%
of companies

Known vulnerabilities in obsolete software were exploited in one-third of all implemented attack vectors.

Most of the exploited vulnerabilities were detected in 2017:

- Remote code execution in Cisco IOS (CVE-2017-3881),
- Remote code execution in VMware vCenter (CVE-2017-5638),
- Remote code execution in Samba (CVE-2017-7494 - Samba Cry),
- Remote code execution in Windows SMB (MS17-010).

Exploits for the majority of vulnerabilities were publicly available (MS17-010, Samba Cry, VMware vCenter CVE-2017-5638), which made the task of exploiting these vulnerabilities much easier.

A common internal network attack was remote code execution via the network service Java RMI and Java class deserialization in Apache Common Collections (ACC) libraries which are used in different products (e.g. in Cisco Lan Management Solution). Deserialization attacks are effective against many software products used by large companies, and help to rapidly gain maximum privileges on critical servers of corporate infrastructure.

Recent vulnerabilities in Windows have been used for remote code execution (MS17-010 Eternal Blue) and local privilege escalation in the system (MS16-075 Rotten Potato). The notorious MS17-010 vulnerability was identified in 60% of all companies and in 75% of companies that underwent penetration testing after information about the vulnerability was published. It should be noted that MS17-010 was detected both in the companies that were tested in late Q1 and in Q2 2017 (detection of the vulnerability was unsurprising because the update had only recently been published), and in companies tested in Q4 2017. The latter fact indicates that update/vulnerability management is not effective enough, and that there is risk of infection by malware such as WannaCry.

Security recommendations:

Monitor publications about new vulnerabilities in software. Update software in a timely manner. Use Endpoint Protection-class solutions with an incorporated IDS/IPS module.

Recommendations for detection:

The following events may be indicative of software vulnerability exploitation attempts and should be watched out for:

- Triggering of the IDS/IPS module in Endpoint Protection-class solutions.
- Server application processes spawning non-typical processes (e.g. Apache web server launches bash or MS SQL launches PowerShell). To monitor such events, process launch events should be collected from end nodes – these events should include information both about the launched process and its parent process. Such events can be sourced from paid-for commercial EDR solutions, from free Sysmon or from the standard Windows audit log starting with Windows 10/Windows 2016. Starting from these versions of Windows, the 4688 event (a new process has been created) contains information about parent process; correlation of PID processes needs to be implemented in earlier versions.
- Incorrect shutdown of client and server software which is typically subject to vulnerability exploitation. Note that this approach has the downside of generating lots of false positives.

Online password guessing attacks

Successfully applied in
58%
of companies

Online password guessing attacks were most often used to gain access to Windows users' accounts and web application administrator accounts.

The password policy rules allow users to choose predictable and easily guessable passwords. Such passwords include: p@SSword1, <Company_name>123 etc.

Usage of default passwords and password reuse facilitated successful password guessing attacks against management interfaces.

Security recommendations:

Implement a strict password policy for all user accounts (user accounts, service accounts, administration accounts of web applications, network device etc.).

Improve user awareness of password protection: choose complex passwords, use different passwords for different systems and accounts.

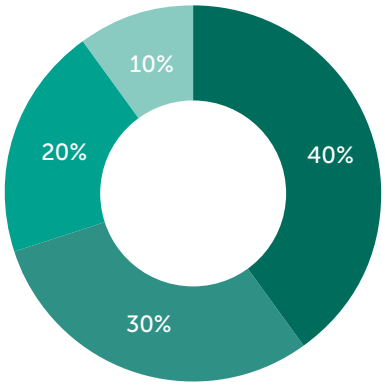
Run an audit of all systems including web applications, CMS and network devices, to check if any default accounts are used.

Recommendations for detection:

To detect password guessing attacks against Windows accounts, attention should be paid to:

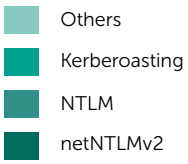
- multiple 4625 events on end hosts (these occur while brute forcing local and domain accounts);
- multiple 4771 events on the domain controller (these occur while brute forcing a domain account using Kerberos);
- multiple 4776 events on the domain controller (these occur while brute forcing a domain account using NTLM).

Offline password guessing attacks



Offline password guessing attacks were launched:

- against NTLM hashes extracted from SAM storage;
- against NetNTLMv2 hashes intercepted during NBNS/LLMNR Spoofing attacks;
- for Kerberoasting attacks (see below);
- against hashes that were obtained from other systems.



Kerberoasting

Weak passwords for accounts with SPN were identified in

20%

of companies

Kerberoasting attack is an offline brute force attack on the password of the account with SPN (Service Principle Name) with which the Kerberos TGS service ticket was encrypted. To launch this type of attack, only domain user rights are required. If the account with SPN has domain administrator rights and the password has been cracked successfully, then the attacker gains access to an account with maximum privileges in the Active Directory domain. In 20% of all companies analyzed, accounts with SPN with weak passwords were identified. In 13% of all companies (or 17% of companies where domain administrator rights were obtained), domain administrator rights were obtained with the help of a Kerberoasting attack.

Security recommendations:

Set a complex password (containing no less than 20 characters) for the account with SPN.

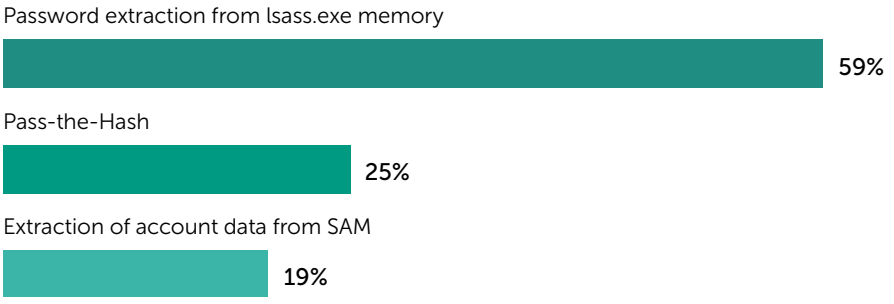
Follow the principle of least privilege for service accounts.

Recommendations for detection:

Monitor requests for a TGS service ticket with RC4 encryption (Windows security log event 4769 with type 0x17). A large number of TGS ticket requests for different SPNs occurring over a small period of time is indicative of an attack.

Kaspersky Lab experts also used a number of Windows network peculiarities which are not vulnerabilities per se, but create a lot of opportunities to carry out so-called lateral movement and to further pursue the attack. The following were used most actively: extraction of users' passwords and hash values from the memory of the lsass.exe process, carrying out a Pass-the-Hash attack, extraction of hash values from the SAM database.

Percentage of attack vectors in which this technique was used



Extracting credentials from the memory of the lsass.exe process

59%
of implemented vectors

Passwords can be obtained because of weak implementation of Single Sign-On (SSO) in Windows systems: some subsystems store passwords in the operating system memory using reversible encoding. A privileged user of the operating system is thereby able to access all logged user credentials.

Security recommendations:

- Follow the **principle of least privilege** in all systems. Besides, it is recommended, whenever possible, to refrain from using local administrator accounts in the domain environment. Follow the Microsoft tier model for privileged accounts to reduce the risks of compromising.
- Use Credential Guard (this security mechanism appeared with Windows 10/Windows Server 2016).
- Use Authentication Policies and Authentication Policy Silos;
- Disable net logon for local administrator accounts or for all members of the 'Local account and member of Administrators group'. (This group appears in Windows 8.1/ Windows Server2012R2, as well as in Windows 7/Windows 8/Windows Server2008R2 with update KB 2871997).
- Use **'Restricted Admin RDP'** instead of using regular RDP. It should be noted that this measure reduces the risk of clear-text password extraction, but increases the risk of unauthorized RDP connections established using the hash value (Pass-the Hash attack). Using this measure is only recommended when a comprehensive approach is observed and measures are taken to protect against Pass-the-Hash attacks.
- Use the **Protected Users** group **for privileged accounts** whose members can only log on via the Kerberos protocol (a list of all protection mechanisms for this group is available on the Microsoft page).
- Enable LSA protection to prevent reading memory and code injection by non-protected processes. This provides added security for the credentials that the LSA stores and manages.
- **Disable WDigest storage** in memory or completely disable use of the WDigest authentication method (applicable to operating systems from Windows 8.1/Windows Server 2012 R2 or Windows 7/Windows Server 2008 with security update KB2871997).
- Disable use of the privilege **SeDebugPrivilege** in the domain policy configurations.
- Disable the **Automatic Restart Sign-On (ARSO)** feature.
- When using privileged accounts for remote access (including via RDP), make sure to log out each time when terminating the session.
- Configure RDP session termination in GPO: Computer Configuration\Policies\ Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Session Time Limits.
- Enable use of SACL for registration of processes that are attempting to gain access to lsass.exe.
- Use antivirus software.

This list of measures does not guarantee complete security. However, it can be used to detect network attacks as well as to reduce the risks of a successful attack, including malware-assisted attacks conducted automatically, such as NotPetya/ ExPetr.

Recommendations for detection:

The approaches to detect attempted password extraction from lsass.exe process memory vary greatly depending on the technique used by the attacker, and lie outside the scope of this publication. For more information, please visit https://kas.pr/16a7.

We also recommend paying extra attention to methods for detecting credential extraction with the use of PowerShell (Invoke-Mimikatz).

Pass-the Hash attack

25%
of implemented vectors

In this type of attack, NTLM hashes obtained from the SAM storage or from lsass.exe process memory are used to authenticate on a remote resource without using the account password.

This attack was successfully used in 25% of attack vectors, affecting 28% of the analyzed companies.

Security recommendations:

The most effective way to protect against this type of attack is to block the use of the NTLM protocol in the network.

Use LAPS (Local Administrator Password Solution) to manage local administrator passwords.

Disable network logons for local administrator accounts or for all members of the 'Local account and member of Administrators group'. (This group appears in Windows 8.1/ Windows Server2012R2, as well as in Windows 7/Windows 8/Windows Server2008R2 with update KB 2871997).

Follow the principle of least privilege in all systems. Follow the Microsoft tier model for privileged accounts to reduce the risks of compromising.

Recommendations for detection:

This attack can be detected most effectively in a well segmented network with strict rules in place for using privileged accounts.

It is recommended to make a list of accounts that may be targeted by attacks. This list should include not only highly privileged accounts but also all accounts that may be used to access an organization's critical resources.

When developing a Pass-the-Hash attack detection strategy, pay attention at non-typical net logon events related to:

- The IP addresses of the source and the target resource;
- The logon times (working hours, vacation dates).

Also, pay attention to non-typical events related to:

- Accounts (creation of accounts, changes to account settings, attempts to use prohibited methods of authentication);
- Simultaneous use of multiple accounts (attempts to log on to different accounts from the same computer, use of different accounts for VPN connection and for access to resources).

Many tools that are used in Pass-the-Hash attacks randomly generate workstation names. This can be detected by 4624 events in which the workstation name is a random combination of characters.

Extracting local user credentials from SAM

19%
of implemented vectors

NTLM hashes of local accounts that were extracted from the Windows SAM storage were used in offline password guessing attacks or in Pass-the-Hash attacks.

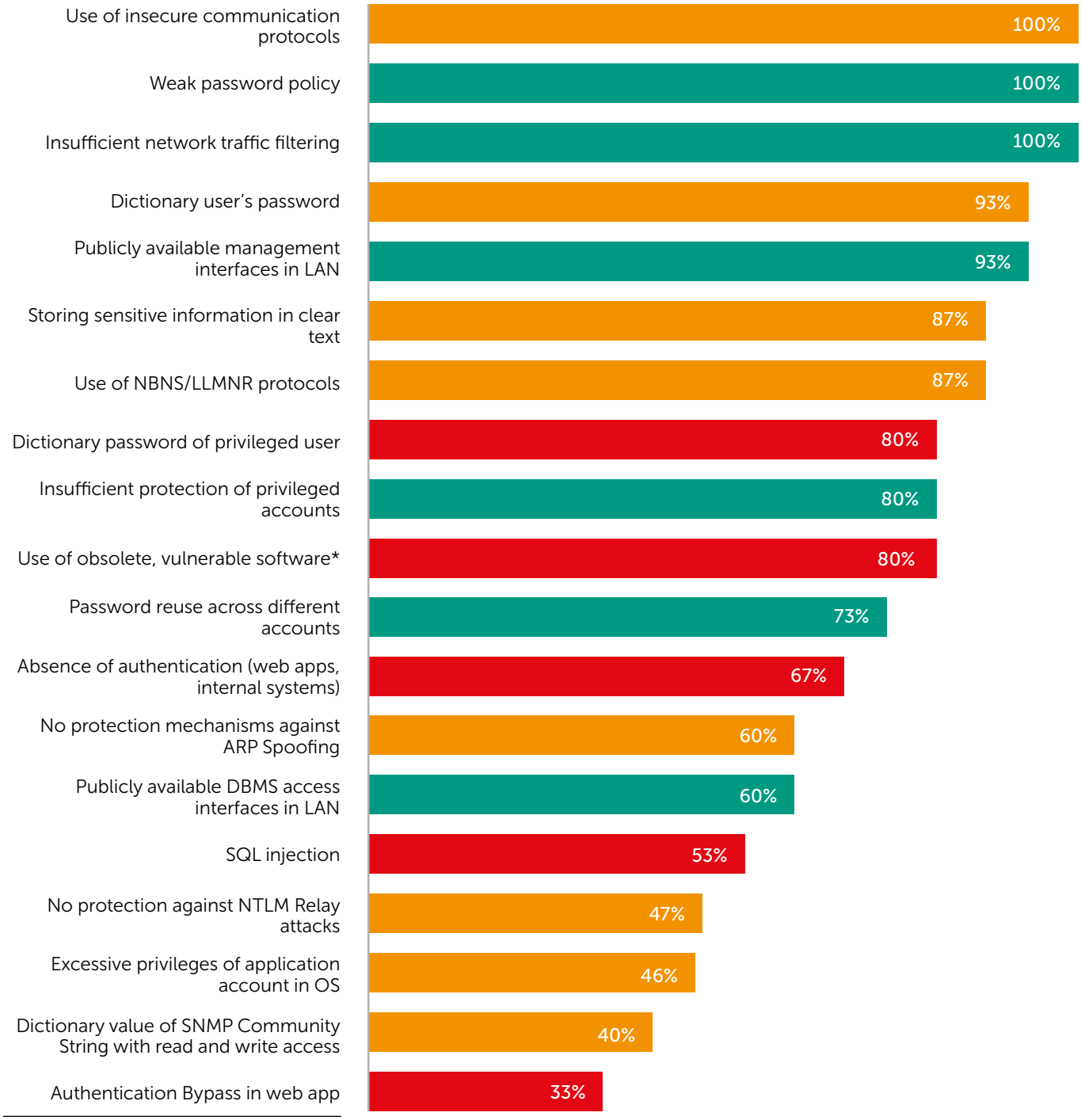
Recommendations for detection:

Detecting attempts to extract login credentials from SAM depends on the method used by the attacker: direct access to the logical volume, Shadow Copy, reg.exe, remote registry, etc.

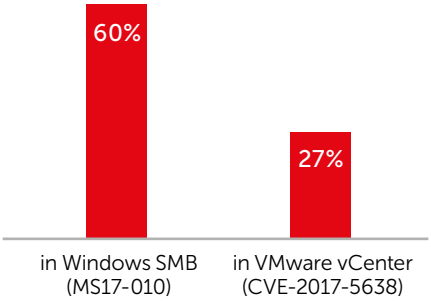
For detailed information on the detection of credential extraction attacks, please visit https://kas.pr/16a7.

Statistics on the most common vulnerabilities and security flaws

The most common vulnerabilities and security flaws



* Remote code execution:



Insufficient network traffic filtering was identified in all analyzed companies. Management interfaces (SSH, Telnet, SNMP, web app management interfaces) and DBMS access interfaces can be accessed from the user segment. Use of weak passwords and password reuse across different accounts made password guessing attacks easier.

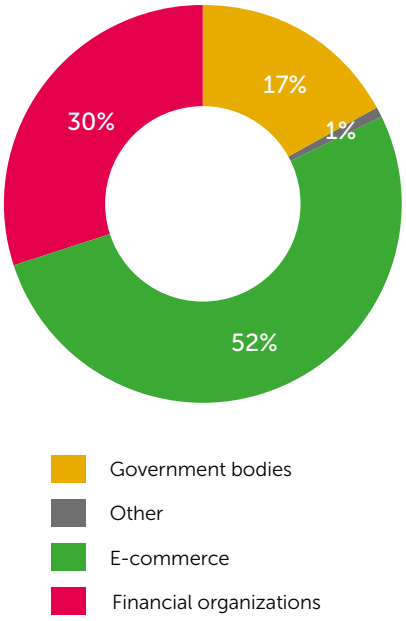
When an application account had excessive privileges in the operating system, exploitation of vulnerabilities in that application made it possible to gain maximum privileges on the appropriate host, and this made the subsequent attack a lot easier.

Web application security assessment

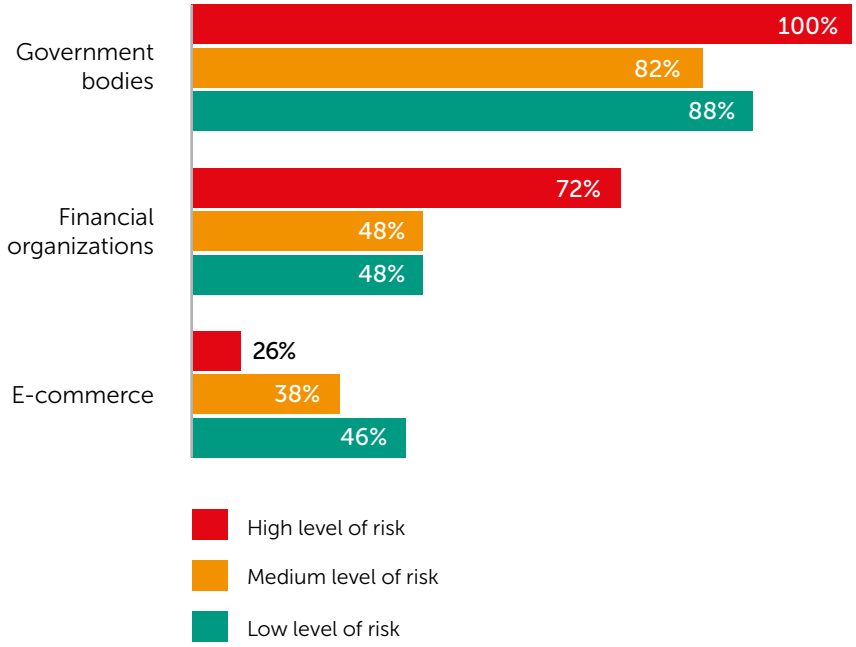
The statistics below include the security assessment results for companies worldwide. 52% of all applications are related to e-commerce.

According to 2017 analysis, applications of government bodies are the most vulnerable, with high-risk vulnerabilities identified in all applications. In e-commerce applications, high-risk vulnerabilities make up the smallest proportion of 26%. The 'Other' category includes just one application, so this category was not taken into account when calculating the statistics for distribution across economic sectors.

Distribution of analyzed applications across economic sectors

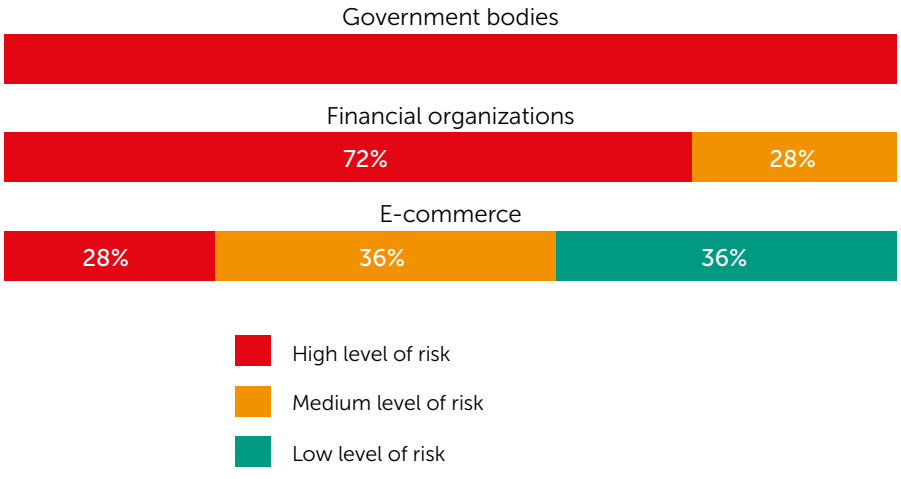


Distribution of analyzed applications by level of risk



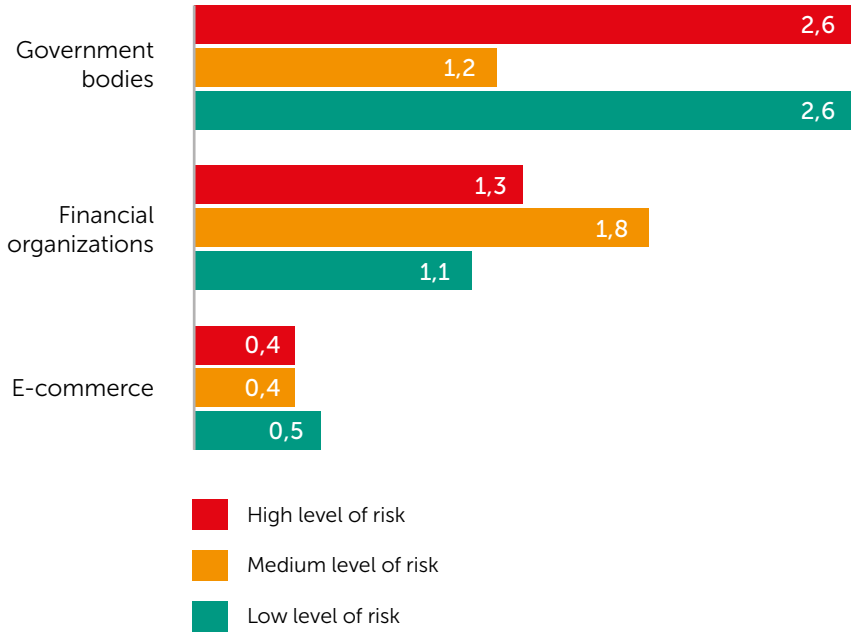
For each application, the overall risk level was assigned based on the maximum risk level of the vulnerabilities detected in it. The applications used in e-commerce were found to be the most secure: only 28% of all applications were found to have high-level risk vulnerabilities, while 36% had vulnerabilities of a medium-risk level at most.

Proportions of vulnerable web applications according to their highest risk levels



If we look at the average number of vulnerabilities per application, then the ranking of economic sectors stays the same: the largest average number of vulnerabilities per app was identified on the websites of government bodies, the second position is taken by financial organizations, followed by e-commerce.

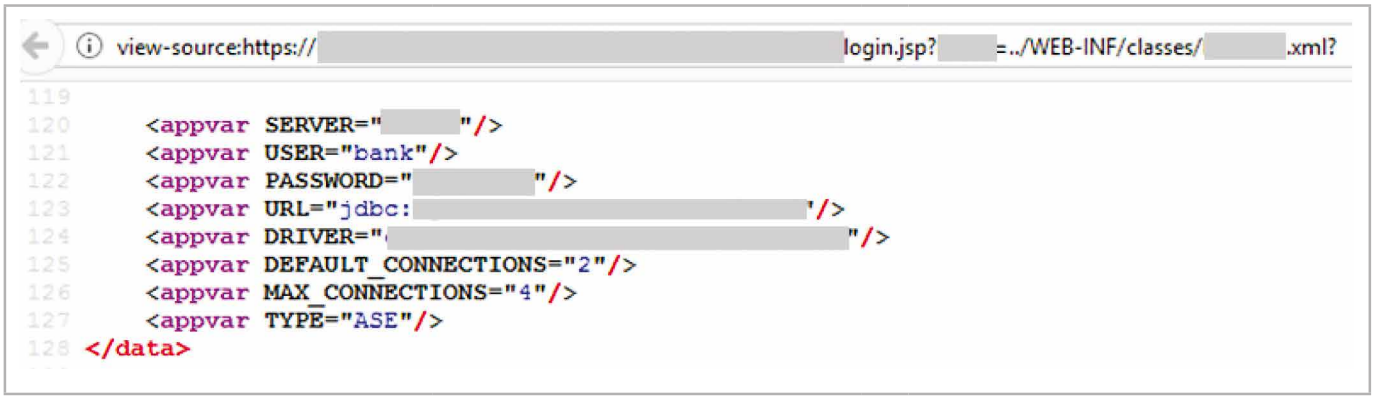
Average number of vulnerabilities per application



In 2017, the largest number of applications had the following types of high-risk vulnerabilities:

- Sensitive Data Exposure (according to OWASP classification), including exposure of web application source code, configuration files, event log files, etc.

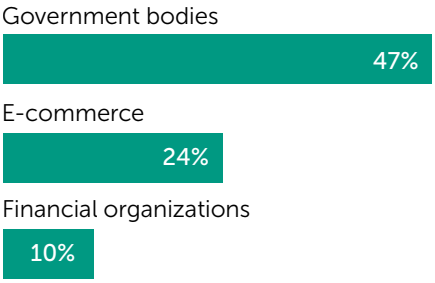
- Unvalidated redirects and forwards (according to OWASP classification). This type of vulnerability typically has a medium risk level and is used to conduct phishing attacks or distribute malware. In 2017, Kaspersky Lab experts typically encountered a more dangerous version of an unvalidated forward-type vulnerability. That vulnerability was present in Java applications and allowed the attacker to carry out path traversal and read various files on the server. In particular, the attacker could gain access to detailed information about users and passwords in clear-text.



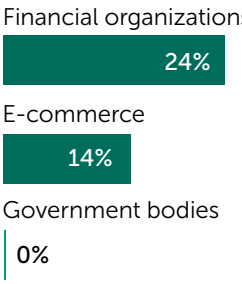
- Dictionary user credentials (vulnerability is included in the Broken Authentication category in the OWASP classification). These were found during online password guessing attacks, offline password guessing attacks with known hash values, and during analysis of web application source code.

In **applications of all economic sectors, sensitive data exposure** (internal IP addresses and ports for database access, passwords, system backup copies, etc.) **and dictionary user credentials were identified**.

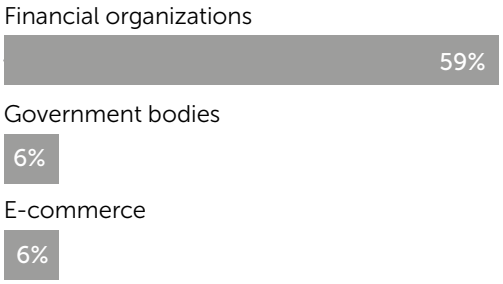
Sensitive data exposure



Unvalidated redirects and forwards

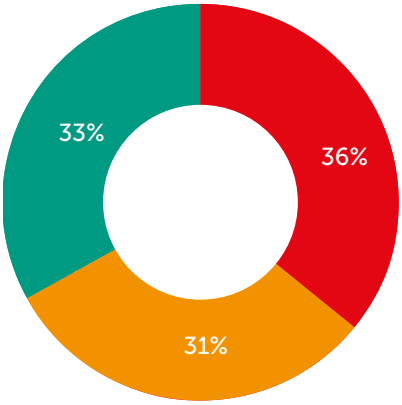


Dictionary user credentials



Vulnerability analysis

Distribution of vulnerabilities by risk level

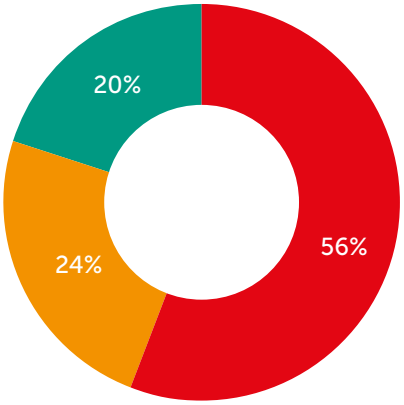


In 2017, roughly equal numbers of high-risk, medium-risk and low-risk vulnerabilities were identified. However, if we look at the overall risk levels of applications, then we see that more than half (56%) of applications contain high-risk vulnerabilities. For each application, the overall risk level was assigned based on the highest risk level of all vulnerabilities found within the application.

More than half of all vulnerabilities are caused by errors in the source code of web applications. The most common of these is cross-site scripting. 44% of vulnerabilities were caused by configuration flaws. The largest number of configuration flaws is related to sensitive data exposure.

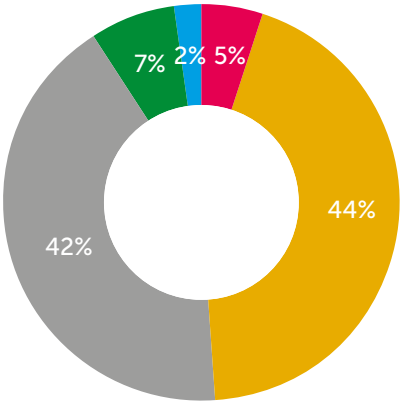
Analysis of vulnerabilities has shown that most of them are related to the server side of applications. Among them, the most common are sensitive data exposure, SQL injection and missing function level access control. 28% of vulnerabilities are related to the client side, more than half of them being cross-site scripting.

Distribution of applications by risk level



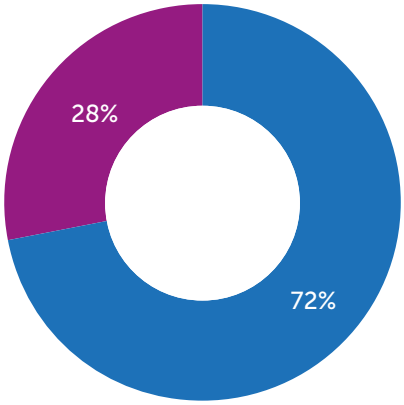
- High level of risk
- Medium level of risk
- Low level of risk

Proportion of different vulnerability types



- Vulnerability in third-party software component (libraries, CMS system plugins, etc.)
- Vulnerability in web server
- Vulnerability in application source code
- Configuration flaw
- Dictionary passwords

Proportions of vulnerabilities on the server side and client side

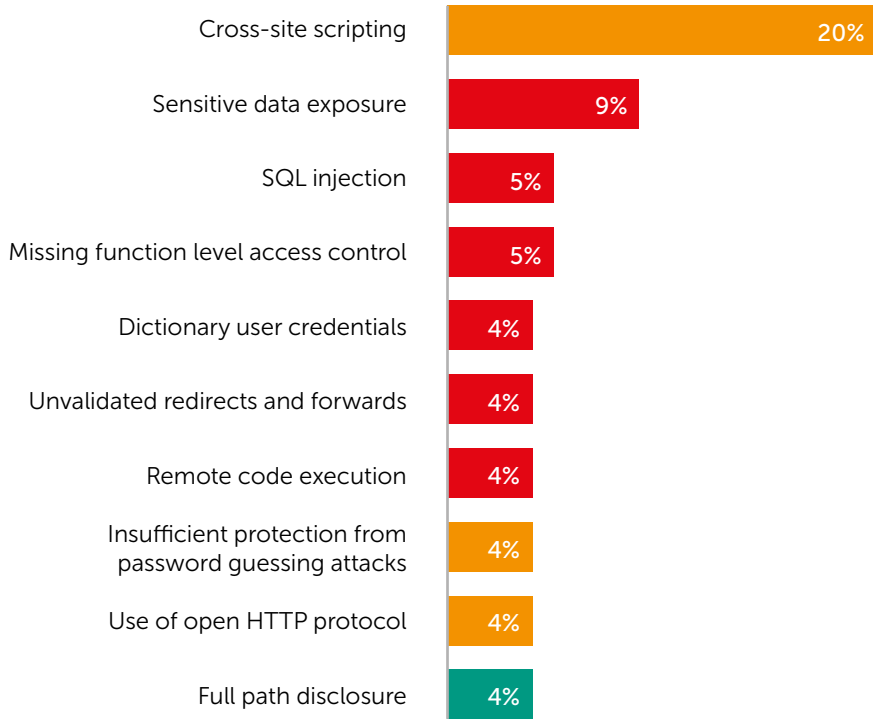


- Server side
- Client side

Statistics on total number of vulnerabilities

This section provides the overall statistics on vulnerabilities. It should be noted that several vulnerabilities of the same type were found in some applications.

The 10 most common types of vulnerabilities



20% of the identified vulnerabilities belong to the **Cross-site scripting** type. The attacker can use this vulnerability to obtain user authentication data (cookies), implement phishing attacks or distribute malware.

Sensitive data exposure – a high-risk vulnerability which is the second most common. It allows an attacker to gain access to an application's sensitive data or user information via debugging scenarios, event logging files, etc.

SQL Injection – the third most common type of vulnerability. It involves the ability to inject SQL operators via the data in an application's user input. If data validation is insufficient, the attacker may alter the logic of the requests sent to the SQL server, and thus obtain arbitrary data from the SQL server, as allowed by the web application's privileges.

In a number of applications, **Missing function level access control** vulnerability is present, meaning users can gain access to application scripts and files that should not be available for their assigned roles. For example, in an application any unauthorized user had access to the web application monitoring page, which could lead to session hijacking, disclosure of sensitive information or service malfunction.

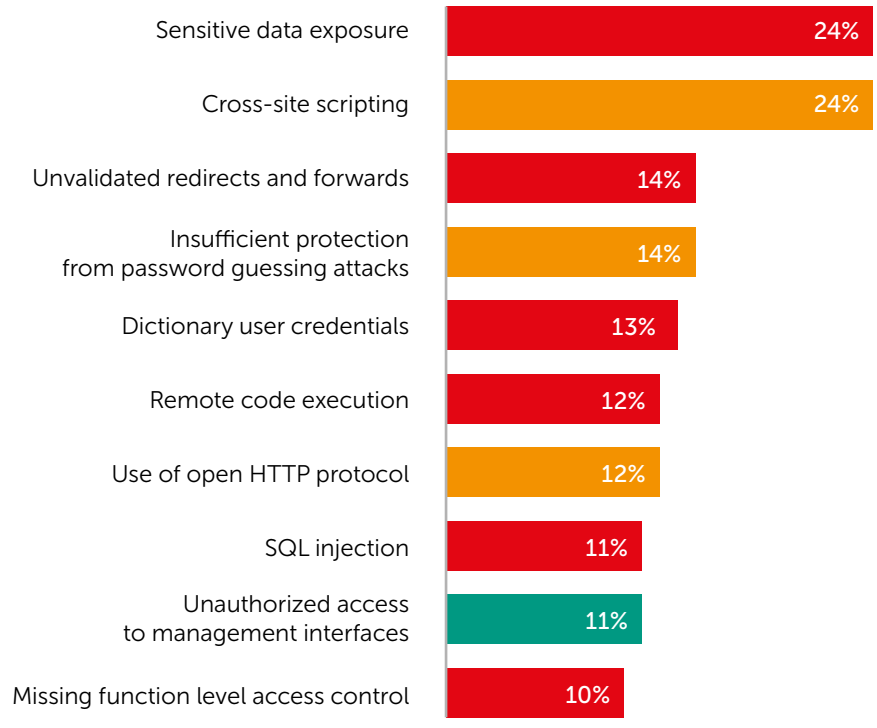
Other types of application vulnerabilities are equally common, each constituting 4% of all identified vulnerabilities:

- **Users use dictionary credentials.** By performing a password guessing attack, the attacker can gain access to the vulnerable system.
- **Unvalidated redirects and forwards** (Unvalidated Forward) allow a remote attacker to redirect the user to arbitrary websites and conduct phishing attacks or distribute malware. In some cases, this vulnerability can be used to gain access to sensitive information.
- **Remote code execution** allows an attacker to execute any commands on the target system or in the target process. This typically involves gaining full access to the application's source code, configuration, access to databases and a chance to pursue the attack further into the network.
- When there is no reliable protection in place against **password guessing attacks** and if the user has a dictionary username and password, an attacker can gain access to the system with the privileges of the targeted user.
- Many applications use the **open HTTP protocol** to transfer data. After implementing a successful man-in-the-middle attack, the attacker can gain access to sensitive data. In particular, if application administrator's credentials are intercepted, the attacker will gain full control over the related hosts.
- **Full path disclosure** in the file system (for the web catalog or other objects of the system) makes other types of attacks easier, such as arbitrary file upload, local file inclusion, and arbitrary file reading.

Statistics for applications

This section provides information about how frequently vulnerabilities occur in applications (the chart below shows the proportion of applications in which each specific type of vulnerability was identified).

Proportion of applications in which most common vulnerabilities were identified



Recommendations for improving web application security

The following measures are recommended to mitigate the risks associated with the above vulnerabilities:

- Check all data coming from web application users.
- Restrict access to management interfaces, sensitive data and directories.
- Follow the principle of least privilege and make sure users have the minimum required permission sets.
- Requirements must be imposed to password minimum length, complexity and password change frequency. The possibility of using dictionary combinations of credentials should be eliminated.
- Updates for software and used components should be installed in a timely manner.
- Implement intrusion detection tools. Consider using WAF. Make sure all preventive protection tools are installed and operate properly.
- Implement Secure Software Development Lifecycle (SSDL).
- Run regular checks to assess the cybersecurity of IT infrastructures, including the cybersecurity of applications.

Conclusion

The overall level of protection against external attackers was assessed as low or extremely low for 43% of all analyzed companies: privileged access to important information systems at these organizations can be gained even by external attackers who are not highly skilled or who have no access to resources except those available to the general public.

Penetration of the network perimeter and gaining access to the internal network was most commonly (73% of attack vectors) carried out via exploitation of vulnerabilities in web applications, such as arbitrary file upload (28%), SQL injection (17%) and others. Another common vector for penetrating the network perimeter was attack on publicly available management interfaces having weak or default credentials and/or via exploitation of vulnerabilities in management interface software. Half of the attack vectors could have been prevented by restricting access to management interfaces (SSH, RDP, SNMP, web management interfaces, etc.).

The level of protection against internal attackers was identified as low or extremely low for 93% of all analyzed companies. Moreover, in 64% of companies more than one vector was identified that could provide the highest privileges in IT infrastructure: Enterprise Admin privileges in Active Directory domain, full control over the network devices and important business systems. On average, 2-3 vectors were identified with which maximum privileges could be gained in each project. It took just three steps on average to gain the domain administrator's privileges in each company.

Both well-known attacks, such as NBNS Spoofing and NTLM Relay, and attacks exploiting vulnerabilities detected in 2017 (MS17-010 (Windows SMB), CVE-2017-7494 (Samba), CVE-2017-5638 (VMware vCenter)) were used to carry out attacks inside internal networks. The vulnerability MS17-010 was detected on internal network hosts in 75% of all analyzed companies that underwent penetration testing after information on the vulnerability was published (MS17-010 is widely exploited during individual targeted attacks as well as malware such as WannaCry and NotPetya/ExPetr that propagates automatically). Obsolete software was identified on the network perimeter of 86% of the analyzed companies and in the internal networks of 80% of companies.

It is worth noting separately the vulnerability of remote code execution via the Java RMI service and via Java deserialization in Apache Commons Collections and in other Java libraries that are used in many out-of-the-box products. In 2017, the OWASP project included the "insecure deserialization" vulnerability in its TOP 10 list of most critical web vulnerabilities (OWASP TOP 10), where it occupied eighth place (A8-Insecure Deserialization). This problem is so common that Oracle is considering the possibility of dropping built-in data serialization/deserialization support altogether in new versions of Java because of the number of vulnerabilities related to these types of operations[1].

Gaining access to a network device often contributes to successful attack development in an internal network. The following vulnerabilities in networking devices were exploited:

- **cisco-sa-20170317-cmp** or CVE-2017-3881 (Cisco IOS), which allows an unauthorized attacker to gain access to the switch with maximum privileges via the Telnet protocol.

- **cisco-sa-20170629-snmp** (Cisco IOS) allows an attacker to gain access to the device with maximum privileges via SNMP protocol knowing only the value of the SNMP Community string with read access (often a dictionary value).

- The Cisco Smart Install feature which is enabled by default in Cisco switches and which does not require authentication. As a result, an unauthorized attacker can obtain and/or replace the switch's configuration file[2].

The 2017 web application security assessment has shown that the applications of government bodies are the most vulnerable (all analyzed applications contained high-risk vulnerabilities), while applications of e-commerce organizations were the least vulnerable (28% applications contained high-risk vulnerabilities). The following types of vulnerabilities occurred most frequently in applications: sensitive data exposure (24%), cross-site scripting (24%), unvalidated redirects and forwards (14%), insufficient protection from password guessing attacks (14%) and dictionary user's credentials (13%).

To improve their security stances, companies are recommended to pay special attention to web application security, timely updates of vulnerable software, password protection and firewalling rules. It is recommended to run regular security assessments for IT-infrastructure (including applications). The task of completely preventing compromising of information resources becomes extremely difficult in large networks, or even impossible when attacks are launched using 0-day vulnerabilities. Therefore, it is important to ensure that information security incidents are detected as early as possible. Timely detection of threat actor activities at the early stages of an attack and a prompt response may help prevent or substantially mitigate the damage caused. Mature organizations where well-established processes are in place for security assessment, vulnerability management and detection of information security incidents, may want to consider running Red Teaming-type tests. Such tests help check how well infrastructures are protected against highly skilled attackers operating with maximum stealth, as well as help train the information security team to identify attacks and react to them in real-world conditions.

1 https://www.bleepingcomputer.com/news/security/oracle-plans-to-drop-java-serialization-support-the-source-of-most-security-bugs/

2 https://dsec.ru/presentations/cisco-smart-install/

Kaspersky Lab

Enterprise Cybersecurity:
www.kaspersky.com/enterprise

Cyber Threats News:
www.securelist.com

#truecybersecurity
#HuMachine

www.kaspersky.com