PT

Penetration testing
of corporate
information systems

External pentests results, 2020

ptsecurity.com

Contents

What is penetration testing

Pentest and pentester

In a penetration test, ethical hackers imitate what real attackers would do. This term is often shortened to "pentest," while the hackers in question are called "pentesters." During a pentest, these pros search for vulnerabilities in the systems of a specific company and attempt to bypass security as part of an attack.

When they are working from an external network (such as the Internet), this is an external pentest. By comparison, in an internal pentest, attacks originate from inside the company (by testing with typical employee privileges or with the physical access available to a random visitor, for example).

In recent years, we have seen a trend towards comprehensive projects in which companies want both external and internal pentesting. Sometimes an internal pentest may be a logical continuation of an external one. This approach allows assessing not just the probability of an attacker penetrating the local network, but also the consequences of developing the attack on company infrastructure.
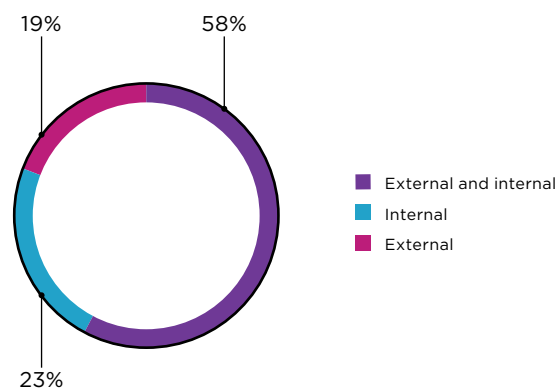

Figure 1. Types of pentests performed in 2019

A pentester is presumed to have the same toolkit and experience as a potential attacker. Logically then, higher levels of skill enable pentesters to better imitate the actions of real-life attackers, and therefore deliver better results.

Of course, unlike real-life attackers, pentesters act strictly within the law and only with the agreement of the system owner. The list of target hosts and tests must be approved in advance by the client.

Why perform pentests

Pentests give a snapshot of the effectiveness of the client's security systems and level of cyberthreat preparedness. If unannounced by management to security staff, such tests can also measure how well the client's security staff detect and disrupt attacks.

Pentesting is not intended to detect vulnerabilities, or in any case, that is not the primary objective. Testers do search for security flaws, but only for the purpose of achieving the objectives of the pentest. In external pentests, the objective is usually to find as many ways to penetrate the local network as possible. The purpose of an internal pentest is to determine the highest level of privileges an attacker can obtain. The client may also set other objectives: for instance, demonstrating access to specific business systems.

Who orders pentests

A pentest can be useful to any organization, irrespective of industry. However, it provides the most value when the client has already secured infrastructure in depth, shored up its cyberdefenses, and deployed security tools. To reach that point, security processes must be sufficiently mature. Pentests are especially important for large companies with geographically diverse infrastructures, due to the sheer difficulty of safeguarding complex systems without testing their security in action.
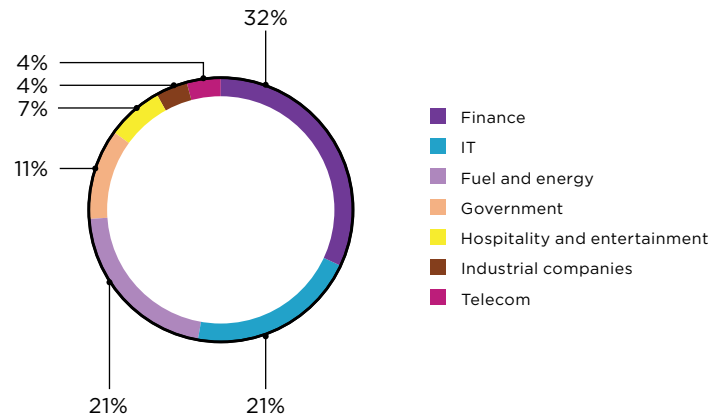

32%

4%
4%
7%

11%

- Finance
- IT
- Fuel and energy
- Government
- Hospitality and entertainment
- Industrial companies
- Telecom

21%   21%

Figure 2. Distribution of tested companies, by industry

What use is pentest for business

Risk assessment and management are key to today's businesses. Executives have a clear understanding of which show-stopping risks must be mitigated no matter what. Many of these risks can occur as the result of a cyberattack: theft of funds from company accounts, say, or losing an important contract due to deletion of files from the CEO's computer.

Management can enumerate these unacceptable risks for pentesters, who then determine whether these risks are in fact plausible and under what circumstances. After this fact-finding stage, recommendations are provided for correctly setting up infrastructure and choosing security solutions to eliminate or mitigate such risks.

There are also other aspects which may be of interest to business, such as:

- Compliance with regulatory requirements and guidelines for information security

- Selection of security tools and points of deployment based on vectors and techniques available to a potential attacker

- Reduced security costs thanks to triage of security flaws and concentrating resources on the most important threats, thereby reducing the risk of system compromise and potential costs of responding to future incidents

- Reputation of the company as a responsible steward of the data of clients and partners

About the research

This report presents the results of external pentesting of corporate information systems performed by Positive Technologies in 2019. Here we describe the most common security issues and attack vectors from our work, as well as recommendations for improving security.

The dataset for this research consists of 28 projects involving external penetration testing for clients consenting to use of such anonymized data for statistical purposes. For accurate and objective results, we only used the most informative projects. To ensure data representative of the true state of security, we have omitted assessments in which pentesting was subject to significant constraints or limited to a small number of hosts.

Only attacks on infrastructure have been included in the report; social engineering and Wi-Fi network attacks are not counted. Results of internal pentesting will be published separately.

Key numbers

- At 93 percent of companies, our pentesters succeeded in breaching the network perimeter and accessing the local network.

- At one sixth of tested companies, they found traces of previous attacks. In other words, the client infrastructure had likely already been under the control of (real) attackers.

- The average time for penetrating a local network was four days. In one case, the time needed was only 30 minutes.

- At 71 percent of companies, even an unskilled hacker would be able to penetrate the internal network.

- Three quarters of penetration vectors (77%) were related to insufficient protection of web applications. Our pentesters discovered at least one such vector at 86 percent of companies.

How we penetrated internal networks

A **penetration vector** refers to a method for exploiting security weaknesses that successfully breaches the network perimeter.

In our 2019 external pentests, we were able to access the local network at 93 percent of tested organizations. Most often, we found several ways of breaching the network perimeter. On average, a single company would have two penetration vectors. The maximum number of penetration vectors detected at a single company was 13.

At one out of every six tested companies, we found traces of prior attacks. For instance, we found web shells on the network perimeter, malicious links on official sites, or valid credentials in public data dumps. This indicates that the infrastructure may have already been under the control of hackers.

Hackers need
from 30 minutes to 10 days
to breach network perimeter

The average time for penetrating a local network was four days. In one case, the time needed was only 30 minutes. In most cases, attack complexity was low, meaning that the attack was within the capabilities of a middling hacker with basic skills. At 71 percent of companies, there was at least one easy penetration vector.

At 68 percent
of companies, an attacker can access the internal network in no more than two steps

Legend:
- 1 step
- 2 steps
- 3 to 6 steps
- Could not breach the perimeter

Figure 3. Minimum number of steps to penetrate the local network (percentage of companies)

An **attack** consists of actions aimed at exploiting a security flaw. An attack can be performed in several successive steps.

An **attack stage** or **step** is an action that allows obtaining data or privileges needed to further develop the attack.
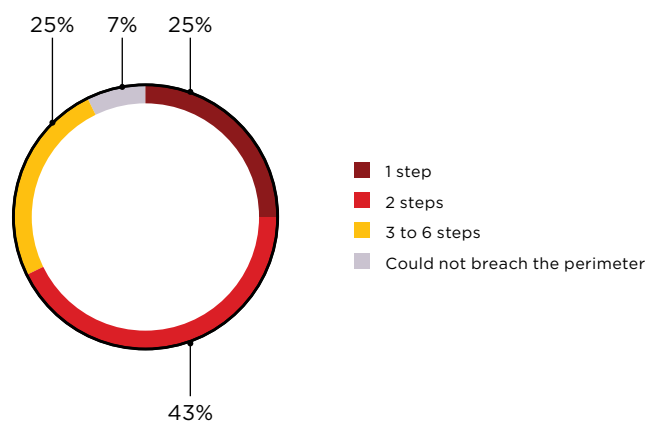
In 77 percent of cases, penetration vectors involved insufficient protection of web applications. At least one such vector was present at 86 percent of companies. The other penetration methods consisted mainly of bruteforcing credentials for services on the network perimeter, including database management systems (DBMS) and remote access.
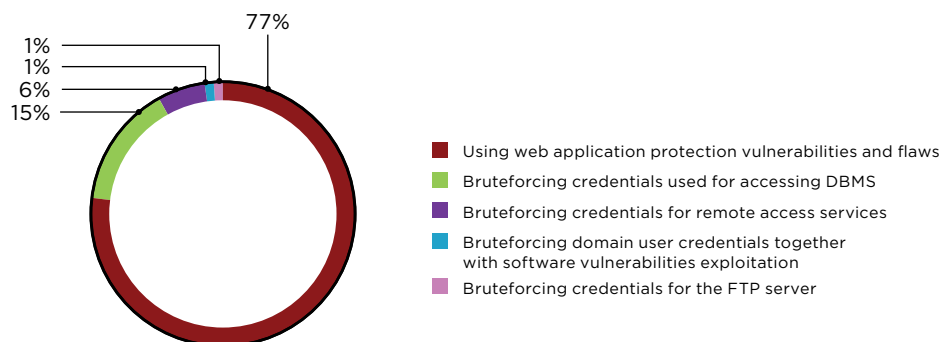
- 77% Using web application protection vulnerabilities and flaws
- 15% Bruteforcing credentials used for accessing DBMS
- 6% Bruteforcing credentials for remote access services
- 1% Bruteforcing domain user credentials together with software vulnerabilities exploitation
- 1% Bruteforcing credentials for the FTP server

Figure 4. Vectors for penetrating the local network (percentage of vectors)

Bruteforcing credentials — 68%
Exploiting vulnerabilities in web application code — 50%
Exploiting known software vulnerabilities — 36%
Using configuration flaws — 29%
Exploiting zero-day vulnerabilities — 14%
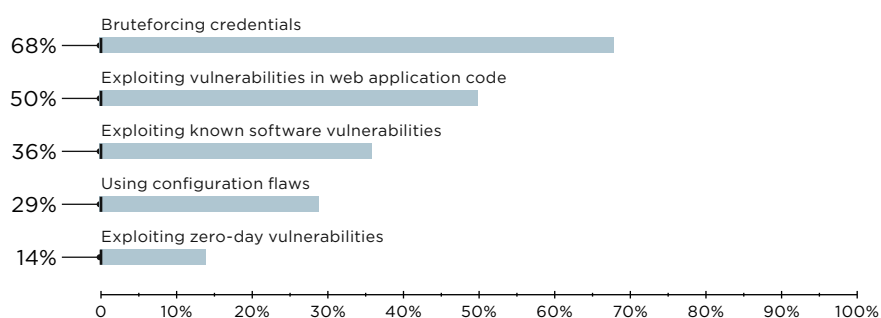
Figure 5. Attacks against web applications leading to penetration of the local network
(percentage of companies)

Recommendations

Perform security assessment of web applications regularly. Because penetration testing is performed as a "black box" without access to source code, it may not be possible to detect some issues. The most thorough testing method is source code analysis (white box), which allows detecting the greatest number of issues. Fixing issues may take the developers significant time. In addition, besides being in in-house web applications, these issues may also appear in third-party software— meaning that the application will remain vulnerable until the third party releases a patch. To protect the network perimeter, we recommend using a web application firewall (WAF) to prevent exploitation of vulnerabilities.

Not every attack ends with a breach of the internal network. However, an attacker may still gain access to other important resources or disrupt business systems. The following diagram demonstrates distribution of successful attacks by category. Most of them were aimed at bruteforcing credentials and exploiting web application vulnerabilities.
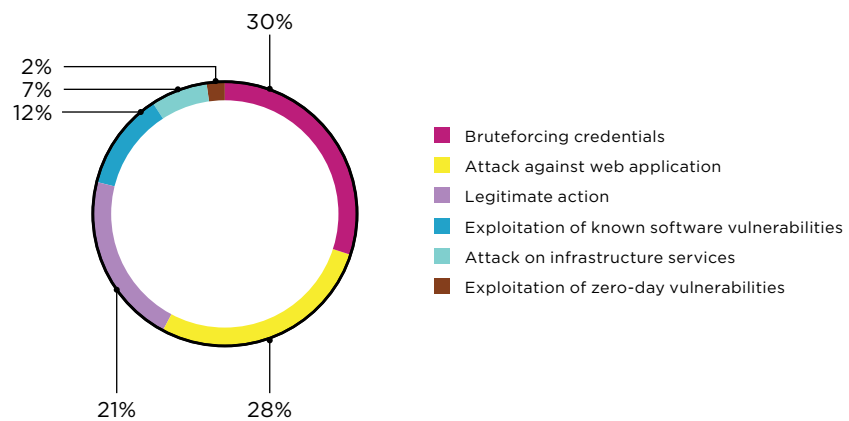
Figure 6. Successful attacks

The following diagram demonstrates the risk level of detected vulnerabilities. Based on CVSS v3.1, each vulnerability is scored as a having a particular level of risk: Critical, High, Medium, or Low. Note that penetration testing was performed using the black-box method. Systems may therefore have contained additional vulnerabilities that could not be detected within the scope of work. Moreover, detecting vulnerabilities was not our objective. Instead, the purpose of pentesting was to provide an objective assessment of the current level of protection against external attacks on a given system.
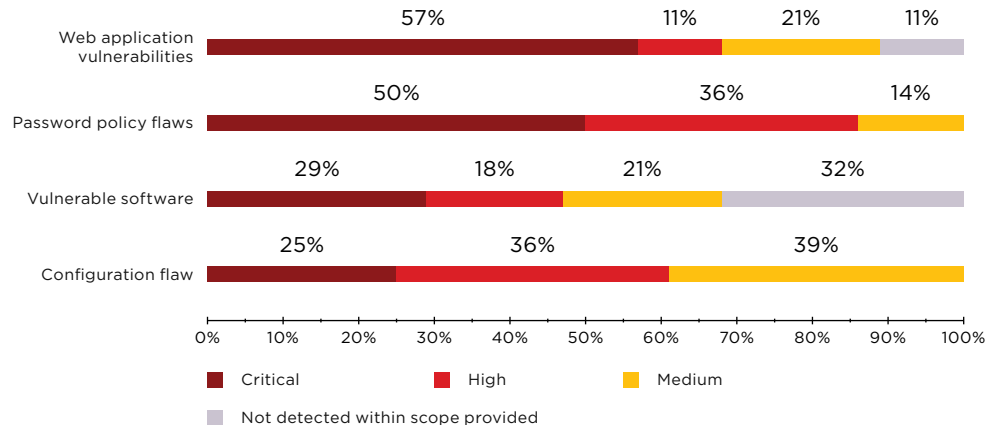
Figure 7. Maximum vulnerability severity (percentage of companies)

An attack on the network perimeter usually starts with bruteforcing user credentials for accessible services, and this step is usually successful.

At 25 percent of companies, identifiers for web applications that use domain authentication were bruteforced via the Autodiscover service in Microsoft Exchange Client Access Server by performing a timing attack. If an identifier exists in the system, the service responds to login attempts within a certain amount of time. This time is usually two seconds, but may vary for different systems. If a particular identifier does not exist in the system, the server response time will be more than two seconds. There is no patch for this flaw. The developer does not find it dangerous and recommends using strong passwords. However, we demonstrated that this flaw can be abused, necessitating that companies deal with the risk of compromise of credentials.

If attackers bruteforce the password for at least one domain account, they can discover identifiers for other users by downloading the Offline Address Book, which lists all email addresses of company employees. At one of the tested organizations, our pentesters obtained over 9,000 email addresses this way.
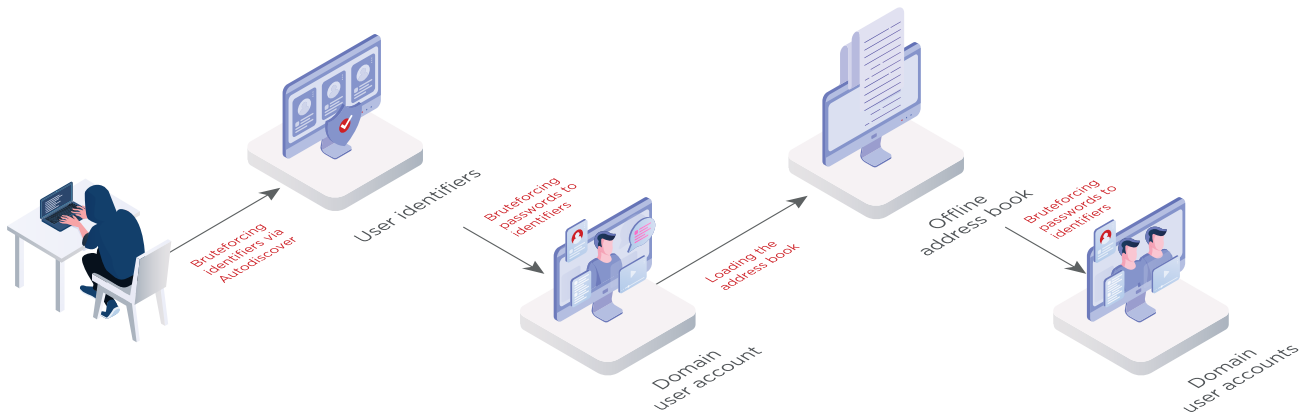

Figure 8. Obtaining accounts of domain users

To obtain identifiers, pentesters also exploited vulnerability CVE-2018-15473 in an old version of OpenSSH. In one such test, they found two penetration vectors at once.

Weak and dictionary user passwords were the main security flaws on the network perimeter. The most popular passwords were those consisting of adjacent characters on the keyboard, such as 123456.
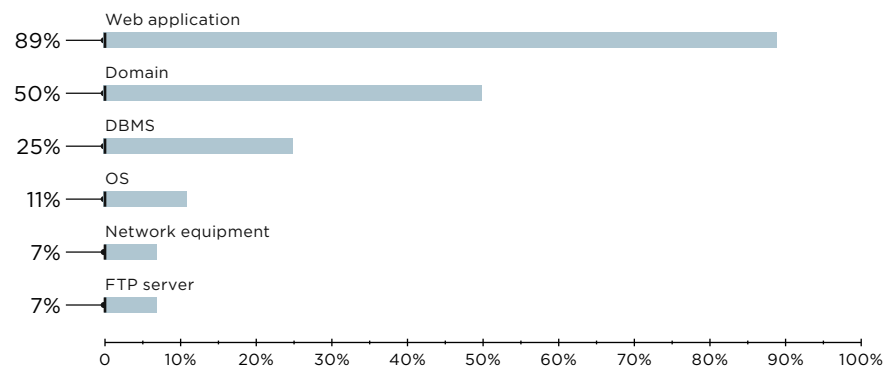
Figure 9. Where we found weak passwords (percentage of companies)
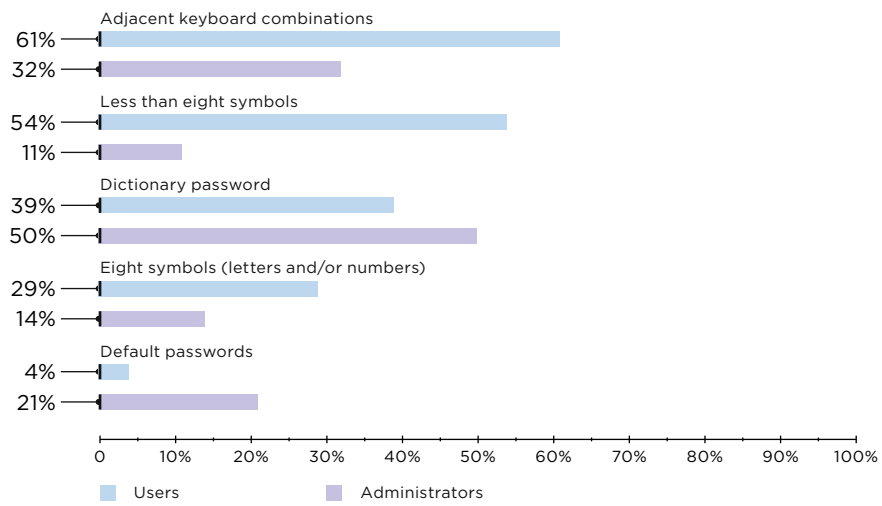
Figure 10. Most common passwords (percentage of companies)

Attack example

Category: Bruteforcing credentials for remote access services

Complexity: Low

After bruteforcing a domain user account, the attacker can connect to remote access services, such as Remote Desktop Service (RDS), as our experts did in one of the tests. The user had access to a limited set of programs, including the 2GIS city map application. The pentesters opened 2GIS help to access the Windows Explorer process and the command line on that host, which allowed them to execute arbitrary OS commands.
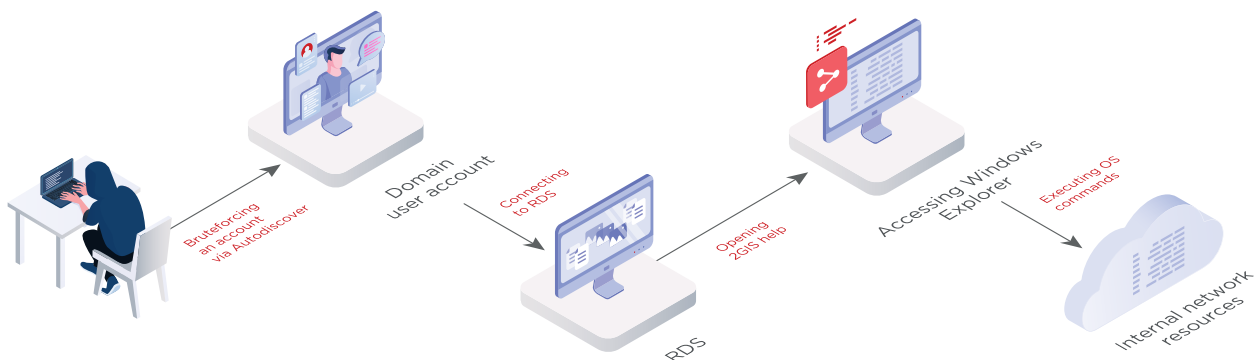
Figure 11. Penetration vector via RDS connection

One third of penetration vectors consisted of just two actions: bruteforcing a web application or DBMS administrator's account, and then executing code via built-in functionality. For instance, PostgreSQL has legitimate functionality for executing OS commands by creating new tables, and the password "postgres" is among the five most common ones.

Attack example

Category: Bruteforcing
Software: pfSense
Complexity: Low

In one pentest, our specialists found that any Internet user could connect to the web management interface of the pfSense firewall by using a default account with the password "pfsense". Functionality built in to the web interface allowed executing OS commands on the server.
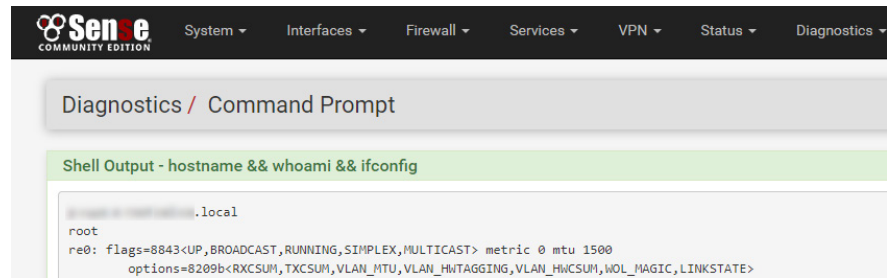
Figure 12. Executing OS commands in the web management interface of the firewall
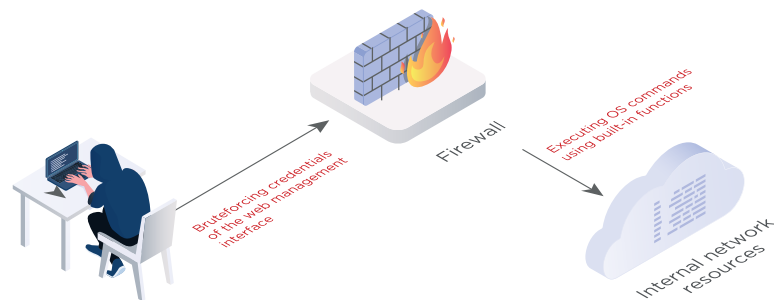

Figure 13. Penetration vector

Recommendations

Make sure that any interfaces open for connection truly need to be available to all Internet users. Regularly take an inventory of the resources that are Internet-accessible.

Forbid use of weak or dictionary passwords. Create and enforce a strict corporate password policy.

Even if a web application does not contain legitimate functionality for command execution, critical errors may still arise made during development or configuration. Here is one example of exploitation of this kind of error.

Attack example

Category: Exploitation of web application vulnerabilities

Complexity: Low

In one application, it was possible to upload documents for antivirus scanning. The administrator could indicate the path to the antivirus engine in the configuration file. That path was replaced with a command for downloading a Perl script. After a document was uploaded by an ordinary user, the application copied the script to the server instead of starting an antivirus scan. Next, the path to the antivirus engine was replaced with a command for executing the script. By uploading another document, the pentesters could connect to the server and execute arbitrary OS commands.
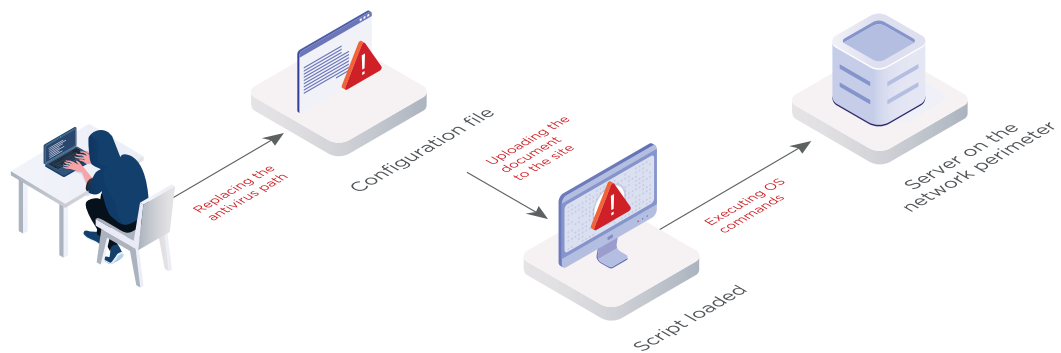
Figure 14. Exploitation of web application vulnerability

To breach the network perimeter, our pentesters widely exploited software vulner-abilities such as CVE-2018-15133 in the Laravel framework, CVE-2018-8284 in .NET Framework, and CVE-2017-10271 in Oracle WebLogic Server. During their work, the pentesters discovered six zero-day Remote Code Execution (RCE) vulnera-bilities, including CVE-2019-19781 in Citrix Application Delivery Controller (ADC) and Citrix Gateway. They found additional zero-day vulnerabilities in other popular products from well-known vendors, but details have been withheld under respon-sible disclosure since patches are still pending.

In early 2020, our experts found two dangerous zero-day vulnerabilities in the Cisco ASA firewall, CVE-2020-3187 and CVE-2020-3259. By exploiting these vul-nerabilities, an attacker could disable the Cisco ASA VPN or access the internal network. Cisco has released security patches, due to which we urge installing cur-rent versions as soon as possible.

Known software security flaws allowed penetrating the local network at 39 per-cent of tested companies. Zero-day vulnerabilities allowed penetration at 14 per-cent of companies.
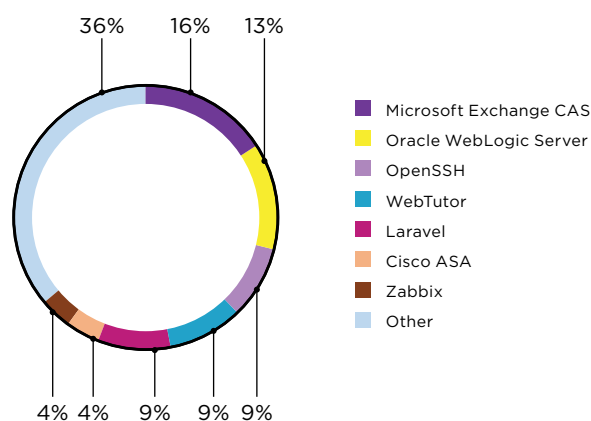

Figure 15. Software in which vulnerabilities were found (percentage of vulnerabilities)
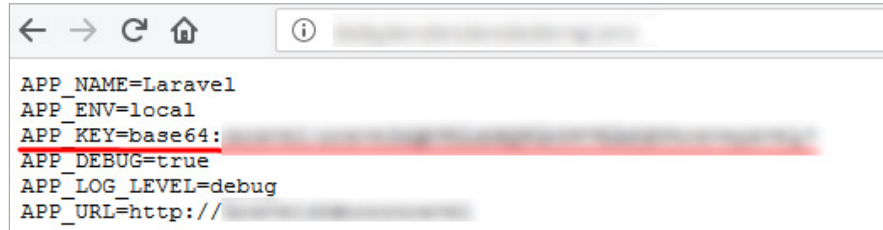
Attack example

Category: Exploitation of known software vulnerability
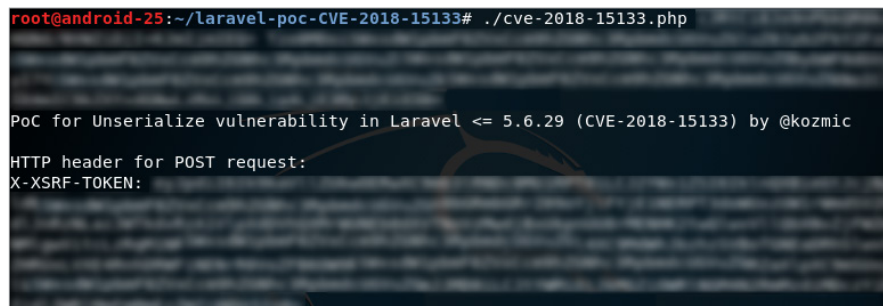
Software: Laravel
Complexity: Low

Here is one example of a local network breach made possible by exploitation of a known vulnerability in the Laravel framework. During testing, our experts found that any external attacker can obtain configuration parameters for the web application environment, including the value of APP_KEY.

Figure 16. Disclosure of web application configuration information

This web application used an outdated version of the Laravel PHP framework that contained a Remote Code Execution vulnerability (CVE-2018-15133). All that is needed to exploit the vulnerability is the value of APP_KEY, which is already known. The pentesters prepared a payload with the help of the PHPGGC utility. When run, the payload establishes a connection from the target host to an external host. The payload was encrypted with APP_KEY and a special public utility. This gave the pentesters the contents of the X-XSRF-TOKEN HTTP header required for the attack. Exploiting the vulnerability yielded access to the internal network.

Figure 17. Creating an HTTP header to exploit the vulnerability

Attack example

Category: Exploitation of known software vulnerability

Software: Microsoft Outlook

Complexity: Low

Let's look at a penetration vector exploiting a vulnerability in an old version of Microsoft Outlook. The attack requires a valid domain user account, which we already know how to obtain.

Microsoft Outlook uses MAPI/HTTP or RPC/HTTP to receive and send emails, as well as store mail processing settings. The Ruler utility allows remotely interacting with a Microsoft Exchange server via these services. If user workstations have obsolete Microsoft Outlook versions installed, an attacker who has a domain user account can create his or her own rules for processing emails, and the rules will be synced with Microsoft Outlook on the client side. The rules may involve running scripts or opening forms that execute VBA code when triggered, such as by receipt of a message with a specific subject line.

During pentesting, our specialists used Ruler and found that MAPI/HTTP and RPC/HTTP were accessible on a Microsoft Exchange server to which domain user accounts had been bruteforced.

```
$ ./ruler-linux64 --domain          --username        --password
  --email                  check
[+] Retrieving MAPI/HTTP info
[+] Binding to RPC
[+] Looks like we are good to go!
$
```

Figure 18. Checking service accessibility with Ruler

For one of the users, the pentesters added an email processing rule to download a .bat file from an external server and run it when a message with a preset subject line is received.

UNC path processing logic in Windows dictates that access to a remote resource will be attempted first via SMB, falling back to WebDAV in case of error. Upon receipt of an email subject with a preset subject line, a WebDAV connection was established, at which point the .bat file was downloaded and run on the user's workstation. This file, in turn, downloads special PowerShell software for connecting to the pentesters' server. As a result, the pentesters were able to execute OS commands with user privileges on the company's local network.
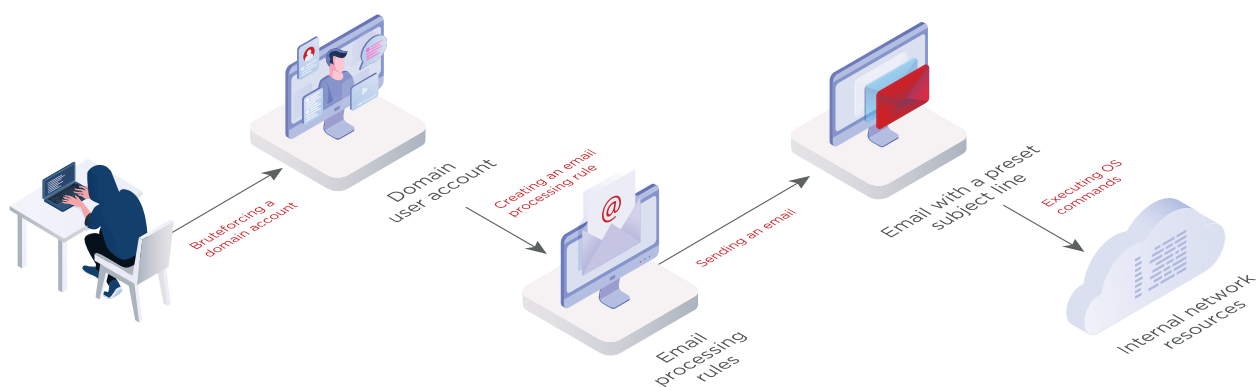

Figure 19. Penetration vector via out-of-date Microsoft Outlook version

Recommendations

Install OS security updates and the latest versions of applications in a timely manner. Make sure that software containing known vulnerabilities does not appear on the corporate network perimeter.

Main threats

Accessing the local network is not the only objective that an attacker may have. Attacks also offer ways to bring about other threats. These could be obtaining control of the company's web application and using it to distribute malware, attacking clients, or bringing down the company's site. Compromise of employee accounts is dangerous: attackers can gain access to resources that rely on domain authentication, such as email. This enables reading confidential correspondence and sending messages posing as company employees or executives. Emails from trusted senders do not raise suspicion, so this attack method is used for fraud, malware distribution, and attacks on other companies. During penetration testing, we merely demonstrate security flaws that would enable such attacks.
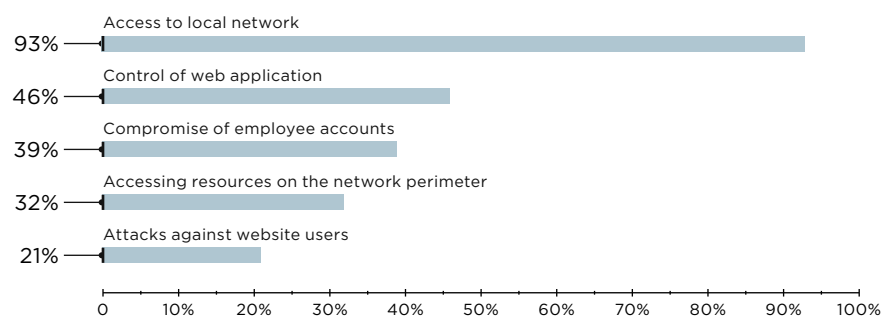
Access to local network
93%

Control of web application
46%

Compromise of employee accounts
39%

Accessing resources on the network perimeter
32%

Attacks against website users
21%

0   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%

Figure 20. Main threats on the network perimeter (percentage of companies)

Conclusions and recommendations

Even an unskilled hacker can penetrate the infrastructure of most tested companies, because many attack vectors involve exploitation of known security flaws. To secure the network perimeter, the first step is to follow basic information security rules. Recommendations for protecting against the most common penetration vectors are given in our research.

Web applications are the most vulnerable component on the network perimeter. Perform security analysis regularly. White-box testing, which includes source code analysis, is the most effective method. Vulnerabilities allowing internal network penetration occur in both in-house apps and solutions by well-known vendors. Fixing them takes time, and meanwhile the application remains vulnerable. For proactive security, we recommend using a web application firewall to prevent exploitation of known vulnerabilities, even ones that have not been detected yet. Usually companies install a WAF only on certain sites. However, keep in mind that WAF solutions can be used to protect many remote access systems. For instance, a correctly installed WAF would stop attackers from exploiting vulnerability CVE-2019-19781 in Citrix Gateway, even before a patch is released and installed.

Penetration testing, regularly performed, detects and closes new penetration vectors. It sheds light on how security at a particular company actually works in practice. And ultimately from a business standpoint, penetration testing examines the plausibility of key business risks related to cyberattacks, providing the basis for an effective and evidence-driven security system.