



Penetration testing of corporate information systems

Internal pentests results, 2020

Contents

About the research	3
Key numbers	3
How we gained control of infrastructure	4
How we accessed business systems	11
Conclusions and recommendations	14

About the research

This report summarizes the results of internal penetration testing performed by Positive Technologies in 2019. We will describe common security flaws and attack methods that can be used by attackers. We will also share our recommendations for improving security.

This builds on our prior research regarding external pentests. During an internal pentest, we simulate attacks from a malefactor located inside the company (for instance, attacks performed with typical employee privileges or as a random visitor).

In an internal pentest, the goal is to determine the maximum level of privileges an attacker can attain. Additionally, the client may set other tasks, such as testing the feasibility of cyberattacks on critical business systems. Penetration testing assesses the effectiveness of the security tools used at the client company. Testing can also measure the preparedness of the client's information security team to detect and prevent attacks, as a sort of unannounced drill.

The dataset here consists of 23 projects involving internal penetration testing for clients consenting to use of such data for statistical purposes. We have included only the most informative projects, so as to provide a more representative assessment of corporate infrastructure security.

Key numbers

- In 2019, we were able to gain full control of infrastructure at all tested companies.
- Gaining control of infrastructure takes up to five days.
- At 61 percent of companies, there is a simple way to gain domain administrator privileges, one that is easy enough even for a poorly skilled hacker.
- Almost half (47%) of all pentester actions blend in with the usual activities of users or administrators, so attacks of the hackers may also go unnoticed.

How we gained control of infrastructure

In 2019, our testers acting as internal attackers managed to obtain full control of infrastructure at all tested companies. Usually it took about three days. One of the networks took just 10 minutes. At 61 percent of the companies, we found at least one simple way to obtain control of infrastructure.

An **attack vector** is a way of gaining access to a target system by exploiting security flaws.

An **attack** refers to malicious actions aimed at exploiting a security flaw. An attack may be performed in several successive steps.

A **stage** or **step** in an attack is an action allowing to obtain data or privileges needed to further develop the attack. In most cases, the number of steps equals the number of vulnerabilities that an attacker needs to successively exploit.

The average attack vector for obtaining domain administrator privileges consisted of six steps. But more steps does not always mean greater complexity. Attack vectors are usually longer on internal networks than on external ones, because the attacker has to move laterally between hosts when searching for the domain administrator account.

Methods of attacking an internal network may involve exploiting software vulnerabilities or OS architecture and authentication mechanisms, as well as performing legitimate actions allowed by the system's functionality.

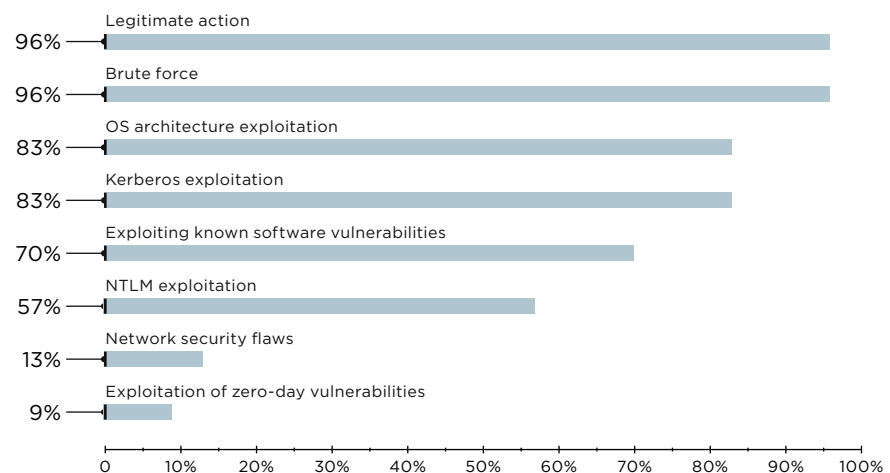


Figure 1. Successful attacks (percentage of companies)

Legitimate actions allowing to develop the attack vector accounted for almost half of all actions performed by the pentesters. These actions included creating new privileged accounts on network hosts, creating a memory dump of lsass.exe, exporting registry hives, sending requests to the domain controller, and cloning virtual machines. It is hard to differentiate between such actions and the usual activities of users and administrators, making it more likely that the attack will remain unnoticed.

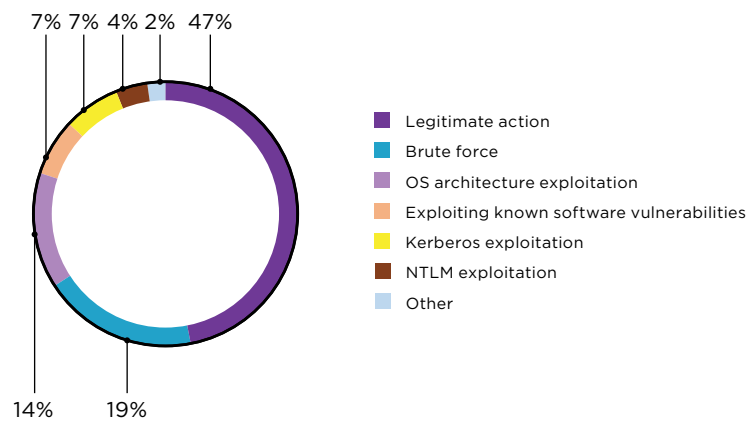


Figure 2. Successful attacks distribution, by category

The most commonly detected vulnerabilities were configuration flaws: unprotected auxiliary protocols, insufficient protection against recovery of credentials from OS memory, and storage of important information in cleartext. Based on CVSS 3.1, each vulnerability is assigned a degree of risk: Critical, High, Medium, or Low. Pentesting is not intended to detect every single vulnerability in a system. Instead, its purpose is to provide an objective and independent assessment of the system's current level of protection against internal attacks.

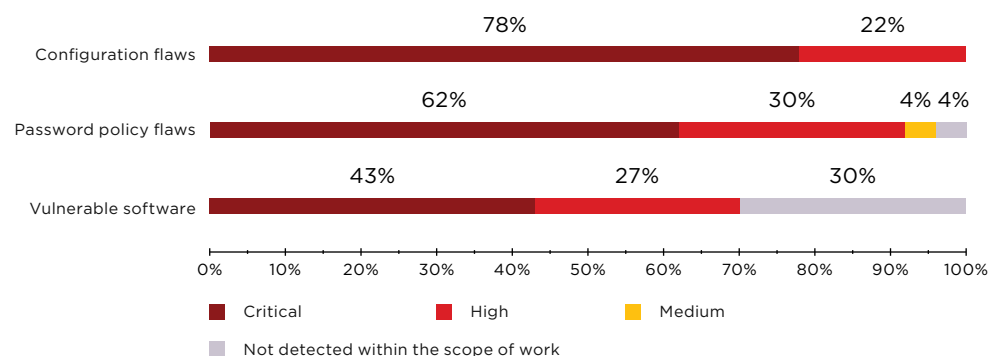


Figure 3. Maximum vulnerability severity (percentage of companies)

Usually the attack vector is based on sequentially obtaining user credentials and moving from host to host until the domain administrator password is discovered. To find the most efficient route, pentesters use Bloodhound software to identify connections between accounts and domain resources, and to determine the groups to which users belong.

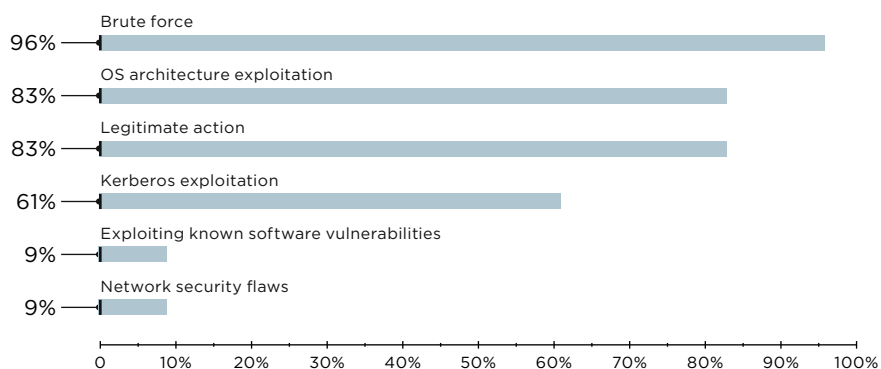


Figure 4. Successful methods of obtaining credentials (percentage of companies)

Peculiarities of the architecture allow extracting credentials from OS memory with special utilities, such as mimikatz and secretdump, or by using built-in OS tools, such as taskmgr, to create a dump of the process lsass.exe. Antivirus protection did not prevent creation of memory dumps. Only in one instance did the client security team receive an alert that procdump had been run.

Recommendations

Protect against recovery of credentials from OS memory. We recommend using Windows 8.1 or later on workstations and Windows Server 2012 R2 on servers. Privileged domain users should be placed in the Protected Users group. In practice, we see that the Protected Users group is rarely used to protect privileged user accounts, although doing so can substantially improve protection against attacks.

Recent versions of Windows 10 and Windows Server 2016 have Remote Credential Guard, which is a technology for isolating and protecting lsass.exe from unauthorized access.

For extra protection of privileged accounts such as domain administrators, we recommend two-factor authentication. On the architectural level, we recommend setting up [an administrative tier model](#) for privileged users.

At 61 percent of companies, our experts successfully used a Kerberoasting attack to take advantage of the Kerberos architecture and obtain credentials. The attack is performed as follows. Any user authenticated on the domain can request a Kerberos ticket for service access (TGS-REP). This request is legitimate. One part of Kerberos TGS-REP tickets is encrypted using the NT hash of the password of the account used to run the service. This password is often weak, or a dictionary password. Service accounts may have administrative privileges, so by performing a dictionary attack using the obtained Kerberos TGS-REP values, attackers can bruteforce passwords and gain access to hosts with such privileges. Passwords are bruteforced on the attacker's computer so security tools can't detect this process. At one of the tested companies, a Kerberoasting attack yielded about 4,000 TGS-REP values, after which 25 passwords for selected privileged accounts were bruteforced.


```

/opt/tools/pentest/impacket/examples$ ./GetUserSPNs.py -request -dc-ip
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

ServicePrincipalName      Name      MemberOf      PasswordLastSet      LastLogon
-----
$krb5tgt$

```

Figure 5. Obtaining TGS-REP values with a Kerberoasting attack

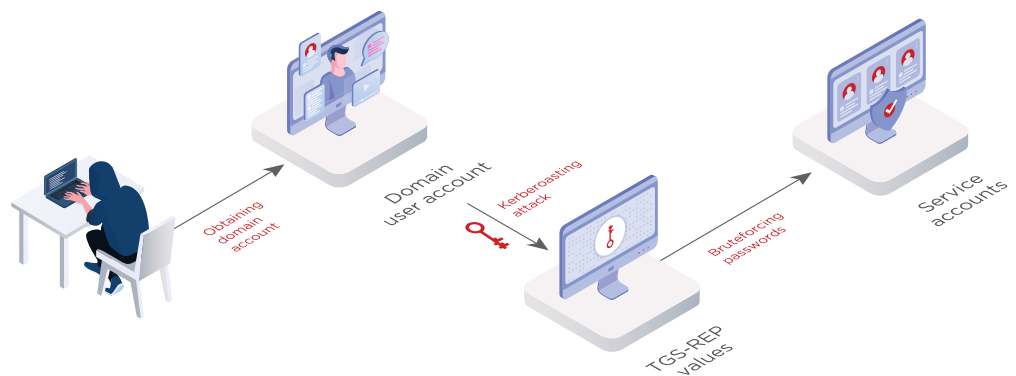


Figure 6. Obtaining credentials for service accounts with a Kerberoasting attack

At many companies, lack of access control allows an arbitrary domain user to obtain local administrator passwords assigned by the Local Administrators Password Service. For example, they can do so by connecting to the domain Active Directory service with Active Directory Explorer.

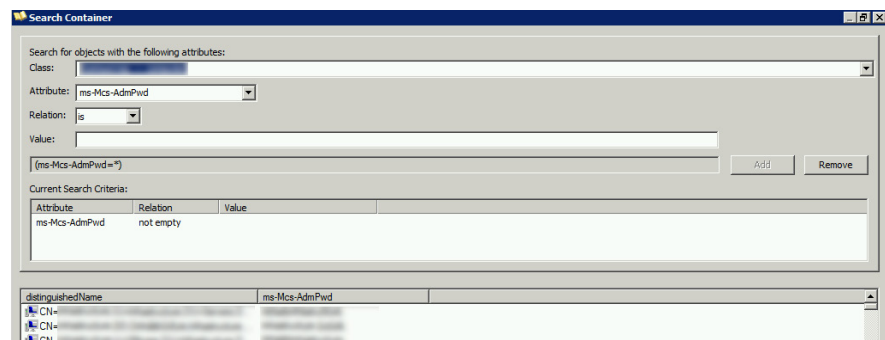


Figure 7. Obtaining local administrator passwords

Recommendations

Restrict the possibility of obtaining credentials from LAPS for non-privileged domain users.

At 94 percent of companies where we performed network traffic analysis, we found various issues with security of auxiliary protocols. However, these protocols are rarely attacked during pentests because doing so may disrupt operations. Network security flaws allow attackers to intercept information transmitted via the network, including credentials. For instance, an attack on the LLMNR and NBNS protocols allows obtaining user identifiers and NetNTLMv2 challenge-response values, which can be used to bruteforce passwords with hashcat.

```
[*] [NBNS] Poisoned answer sent to 10.answanswer for name (service: Workstation/Redirector)
[*] [LLMNR] Poisoned answer sent to 10.answer.swer for name
[Proxy-Auth] NTLMv2 Client :
[Proxy-Auth] NTLMv2 Username :
[Proxy-Auth] NTLMv2 Hash :
```

Figure 8. Intercepting NetNTLMv2 challenge-response values with Responder

Recommendations

Disable NBNS and LLMNR, if possible. If NBNS is necessary for normal operation, use static entries for the main network hosts (such as the default gateway and servers) and use a WINS server instead of broadcast requests. Be careful if using LLMNR. Do not set it as the primary mechanism for resolving IP addresses. Restrict its area of application.

In almost every project, we were able to bruteforce user passwords, including with related data such as NT hashes and TGS-REP values. Even among privileged users, the most common passwords contained neighboring keyboard symbols, such as Qwerty123.

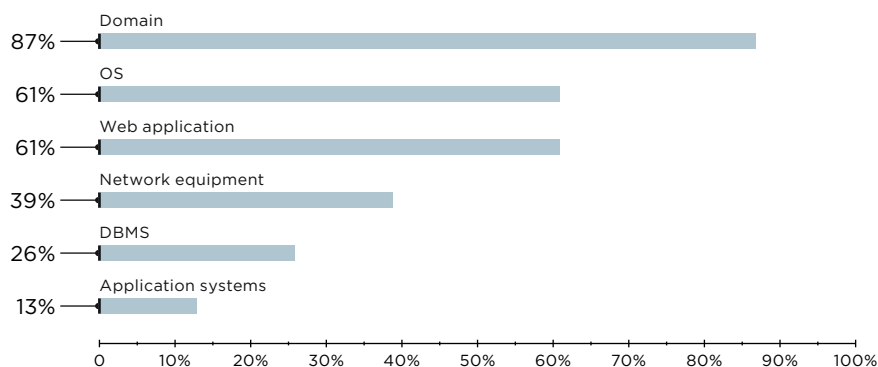


Figure 9. Locations where weak passwords were found (percentage of companies)

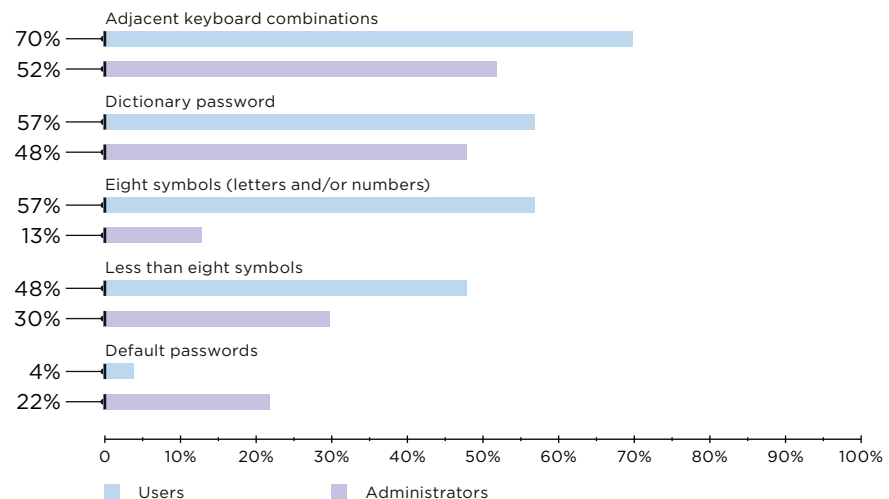


Figure 10. Most common types of passwords (percentage of companies)

Recommendations

Prevent use of dictionary and other easy-to-guess passwords. Develop and enforce a strict corporate password policy.

Domain administrators can check for use of dictionary passwords by exporting the file `ntds.dit` from the controller. This file stores password hashes of accounts for all domain users. One can try to perform an offline dictionary attack on the user passwords in that file. If the attack is successful, talk to the relevant users and explain why they need to set stronger passwords. This should be done periodically.

In addition, consider using published lists of frequently used passwords to forbid use of such passwords on the domain.

The architecture of NTLM and Kerberos is conducive to lateral movement between network hosts. Such techniques include pass-the-hash, NTLM relay, Kerberos silver ticket, and Kerberos golden ticket.

Attackers can exploit known vulnerabilities found in outdated software versions to remotely execute arbitrary code, escalate privileges, or learn important information, such as with [CVE-2019-2725](#) in Oracle WebLogic, [CVE-2019-0686](#) in Microsoft Exchange Server, and [CVE-2018-9276](#) in PRTG Network Monitor. But what we see most often is lack of current OS updates. For example, in Windows we encountered [CVE-2019-0708](#) (BlueKeep) and [CVE-2019-1040](#). At some companies, we still find vulnerabilities described in security bulletin [MS17-010](#) (30 percent of tested companies) and even [MS08-067](#). During penetration testing, we also discovered two zero-day vulnerabilities.

At 22 percent of tested companies, we obtained not just domain administrator privileges, but those of domain forest admin (Enterprise Administrator). Among other things, this required performing SID-History Injection. The attack is performed as follows. If a trust relationship exists between child.domain.local and domain.local, an attacker with maximum privileges on child.domain.local can attempt to obtain maximum privileges on the root domain domain.local.

Domain trusts

CN	flatName	securityIdentifier	trustAttributes	trustDirection	trustType
		S-1-	WITHIN_FOREST	BIDIRECTIONAL	UPLEVEL, MIT
		S-1-	FOREST_TRANSITIVE	BIDIRECTIONAL	UPLEVEL, MIT
		S-1-	WITHIN_FOREST	BIDIRECTIONAL	UPLEVEL, MIT
		S-1-	WITHIN_FOREST	BIDIRECTIONAL	UPLEVEL, MIT
		S-1-	FOREST_TRANSITIVE	BIDIRECTIONAL	UPLEVEL, MIT

Figure 11. Domain trust information

The attack relies on modification of the `sidHistory` attribute. The attribute is intended for easier domain migration, for instance when companies merge or when the domain of a subsidiary is incorporated into the parent company domain. When the old domain `old.local` and the new domain `new.local` need to be merged into a single domain infrastructure, all objects on `old.local` get new security identifiers (SIDs) on `new.local` and their old SIDs are written to `sidHistory`. After the domains are merged, both `objectSid` and `sidHistory` are checked during authentication. The field with the `sidHistory` attribute is not filtered during cross-domain authentication for domains in a trust relationship.

Kerberos packets have a field called `ExtraSids`, which contains the value of `sidHistory`. For the attack, one needs to learn the username of the interdomain trust account (with the `LDAPPER` utility, for instance) and get the NT hash of its password. The password hash is then used to create an Inter-Realm Ticket Granting Ticket (Inter-Realm TGT) with the `ticketer` utility. The root domain SID value is indicated in the `ExtraSids` field. The Inter-Realm TGT is used to request a Kerberos Silver Ticket for the root domain controller.

```
$ python2 /opt/impacket/examples/ticketer.py -domain -domain-sid S-
-nthash -extra-sid S-
-500 -spn 'krbtgt/
Impacket v0.9.18-dev - Copyright 2018 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for
[*] PAC_LOGON_INFO
[*] PAC_CLIENT_INFO_TYPE
[*] EncTicketPart
[*] EncTGSRepPart
[*] Signing/Encrypting final ticket
[*] PAC_SERVER_CHECKSUM
[*] PAC_PRIVSVR_CHECKSUM
[*] EncTicketPart
[*] EncTGSRepPart
[*] Saving ticket in ccache
$
```

Figure 12. Creating Inter-Realm TGT

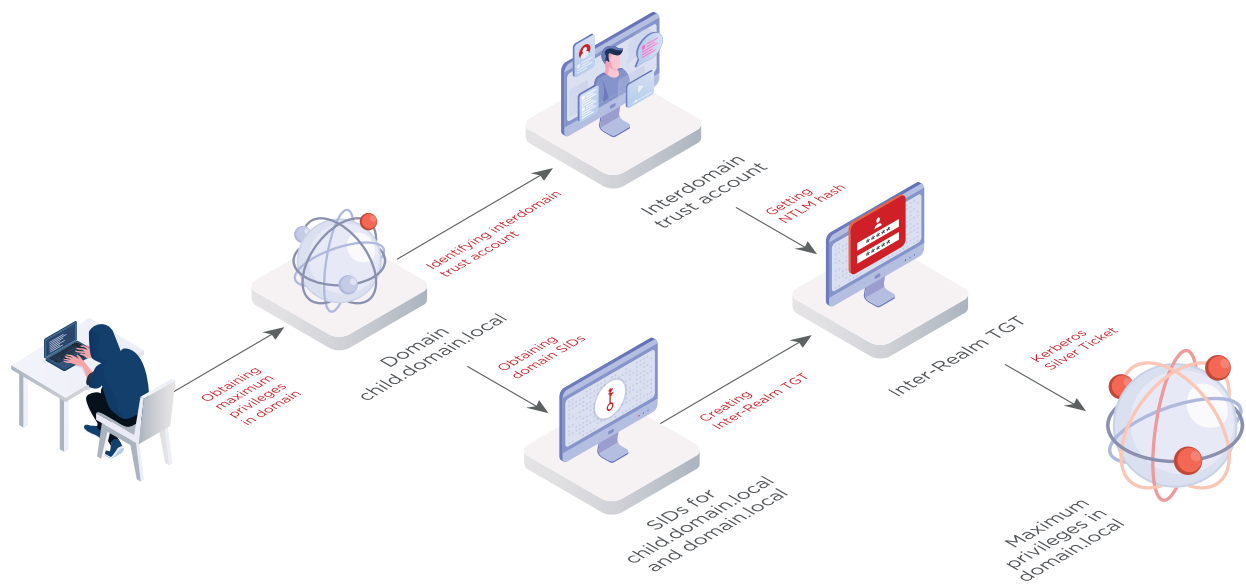


Figure 13. Obtaining maximum privileges on the root domain

Recommendations

Disable sIDHistory for the forest of domains in a trust relationship. SID Filter Quarantining is also an option for domains in a trust relationship (with the TREAT_AS_EXTERNAL attribute). We recommend taking untrusted domains out of the forest first, and then setting up SID filtering for them.

How we accessed business systems

As part of internal pentesting, testers may be asked to demonstrate the feasibility of actuating business risks or obtaining access to business systems. Risks vary by company, but some of them are common to many, such as compromise of critical information in case of access to executive workstations. Here are a few examples of how an attack vector can be developed.

Domain administrator privileges enable performing a Kerberos Golden Ticket attack. The Kerberos protocol works by providing tickets confirming the level of user access to domain infrastructure resources. Privileges for the krbtgt account make it possible to issue a ticket with any level of access. So an attacker who has the NT hash of krbtgt, including a domain administrator, can create a Golden Ticket for accessing resources with any level of privileges.

To deal with the consequences of a Kerberos Golden Ticket attack, the password for the krbtgt account must be changed twice, the incident needs to be investigated, and systems on the compromised hosts must be reinstalled.

With a Golden Ticket attack, hackers can connect to the computer of an executive, install software for hidden remote access such as a modified version of TeamViewer or VNC, and secretly monitor every action of the user.

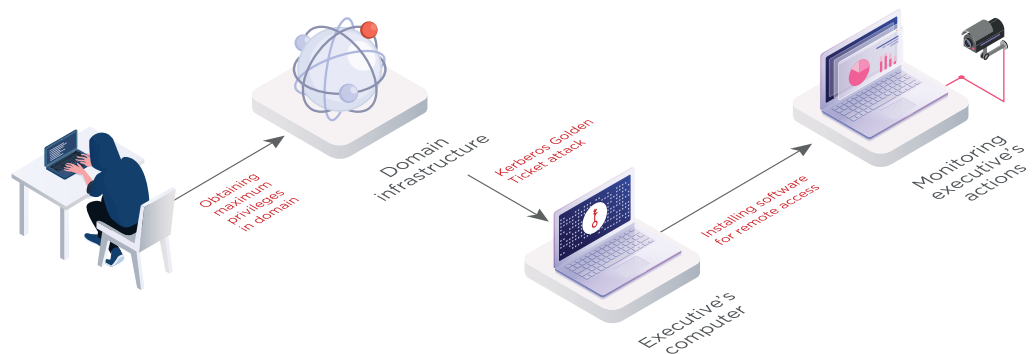


Figure 14. Attack vector for accessing an executive's workstation

Let's consider another example. During an internal pentest, we determined the address of the terminal server from which employees connected to TranzWare Online, which they use to manage an ATM network. Because the pentesters already had domain administrator privileges, they could perform a Golden Ticket attack for accessing the server. Next, they created a memory dump of lsass.exe and used mimikatz to extract user passwords in cleartext. When our specialists connected to the server with the obtained credentials, they found that access to TranzWare Online uses the same passwords as the domain accounts. They then demonstrated that authorization in the system with these credentials was possible. With such an attack, a hacker could reconfigure ATMs, view the transaction list and other reports on the ATM, and attempt to perform fraudulent operations and steal cash.

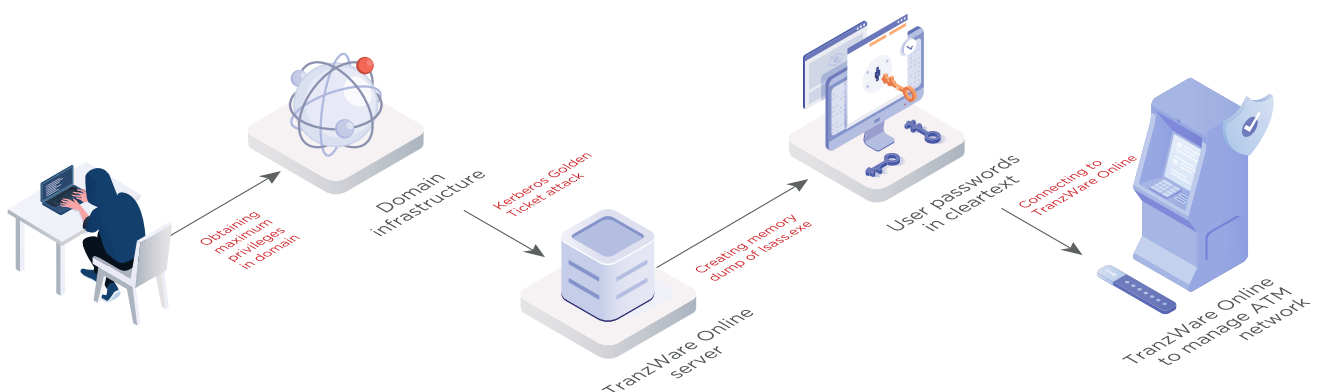


Figure 15. Vector of attack on an ATM network

Manufacturing, industrial, fuel, and energy companies in particular must check the security of their industrial networks. In practice, attackers are able to slip from corporate IT networks onto industrial networks containing sensitive industrial control system (ICS) components. Here is one such attack vector.

During a pentest, our specialists created a memory dump of lsass.exe on one of the network hosts and extracted several accounts from it. Analysis of domain infrastructure with Bloodhound showed that one of the accounts has local administrator privileges on many of the network hosts. The pentesters gathered employee-related information from the corporate portal and successfully used the privileged account to connect to the workstations of users who would reasonably have access to industrial control systems. On one of the workstations, the specialists found information about the industrial network, connection methods, and credentials.

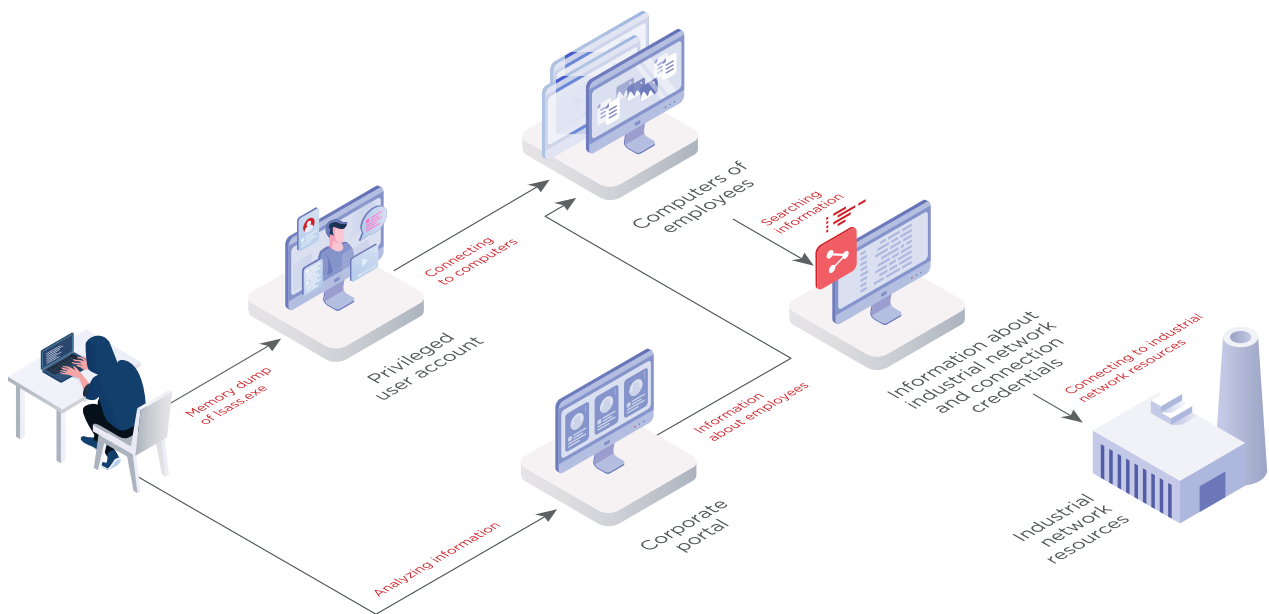


Figure 16. Vector of attack on industrial network

Conclusions and recommendations

Consistent with past years, penetration testing of corporate information systems demonstrates a low level of protection from an internal attacker, who therefore can obtain full control of infrastructure and access critical business systems. Most attacks are possible because of configuration flaws and weak password policies. As such, the very first step for companies is to follow basic security rules. This report provides recommendations for protection from common attack methods.

Since the actions of our pentesters were rarely noticed, actual attackers could presumably persist on infrastructure for long periods as well. For instance, during external penetration testing we found traces of previous attacks that had not been detected by security teams at the time. It is important to use modern tools for early detection of attempted attacks. Security Information and Event Management (SIEM) solutions allow detecting suspicious activity and taking measures to prevent attacks and minimize the consequences. In addition, we recommend regular retrospective event analysis and analysis of traffic inside the network to detect attackers' presence when a system has been hacked. Deep network traffic analysis indicates the fact of compromise and also enables spotting the actions of attackers on target infrastructure. Timely detection is essential for preventing devastating consequences such as theft of confidential information, disruption of business processes, and financial losses.

We recommend performing penetration tests on a regular basis for ground-truth evaluation of the information security measures in place. By empirically assessing anticipated business risks, penetration testing enables building an efficient, effective security system based on the best available options.

About Positive Technologies

ptsecurity.com
pt@ptsecurity.com
facebook.com/PositiveTechnologies
facebook.com/PHDays

For 18 years, Positive Technologies has been creating innovative solutions for information security. We develop products and services to detect, verify, and neutralize the real-world business risks associated with corporate IT infrastructure. Our technologies are backed by years of research experience and the expertise of world-class cybersecurity experts.

Over 2,000 companies in 30 countries trust us to keep them safe.

Follow us on social media ([LinkedIn](#), [Twitter](#)) and the [News](#) section at ptsecurity.com.