IT-Sicherheit

- Setup Virtual Machines
- Härtung Firewall
- Weiteres
- Kali Linux
- Härtung Suricata [SQL-Injection]
- Penetration Testing

Setup Virtual Machines

- Netzwerkkarte 1: Bridge - automatische Erkennung
- Netzwerkkarte 2: Privat auf meinem Mac
- net-tools auf allen installieren
- tcpdump auf allen installieren
- iptables auf allen installieren
- iptables-persistent auf allen installieren

```
apt update && apt upgrade -y


apt install net-tools tcpdump iptables iptables-persistent -y
```

1. **Firewall erstellen**
   1. IP: 10.10.10.2
   2. ssh installieren
3. **Webserver erstellen**
   1. IP: 10.10.10.3
   2. apache2 installieren
4. **Adminserver erstellen**
   1. IP: 10.10.10.4
   2. ssh installieren

Nach Software-Installation die Interfaces ins Internet auf Webserver und Admin deaktivieren!

Netzwerkkonfiguration (/etc/network/interfaces)

```
# Webserver


auto ens256
iface ens256 inet static
  address 10.10.10.3
  gateway 10.10.10.2
  netmask 255.255.255.0


# Firewall
```

```
auto ens256
iface ens256 inet static
  address 10.10.10.2
  netmask 255.255.255.0

# Adminserver

auto ens256
iface ens256 inet static
  address 10.10.10.4
  gateway 10.10.10.2
  netmask 255.255.255.0
```

Firewall-Regeln

Firewall-Maschine:

```
iptables -A PREROUTING -t nat -i ens256 -p tcp --dport 80 -j DNAT --to 10.10.10.3:80

iptables -A PREROUTING -t nat -i ens256 -p tcp --dport 22 -j DNAT --to 10.10.10.4:22
```

Commands

Enable ip-forwarding (auf der Firewall):

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Persistieren vom ip-forwarding: In Datei /etc/sysctl.conf

netstat:

```
netstat -rn
```

tcpdump:

```
tcpdump -i <INTERFACE> port <PORT>
```

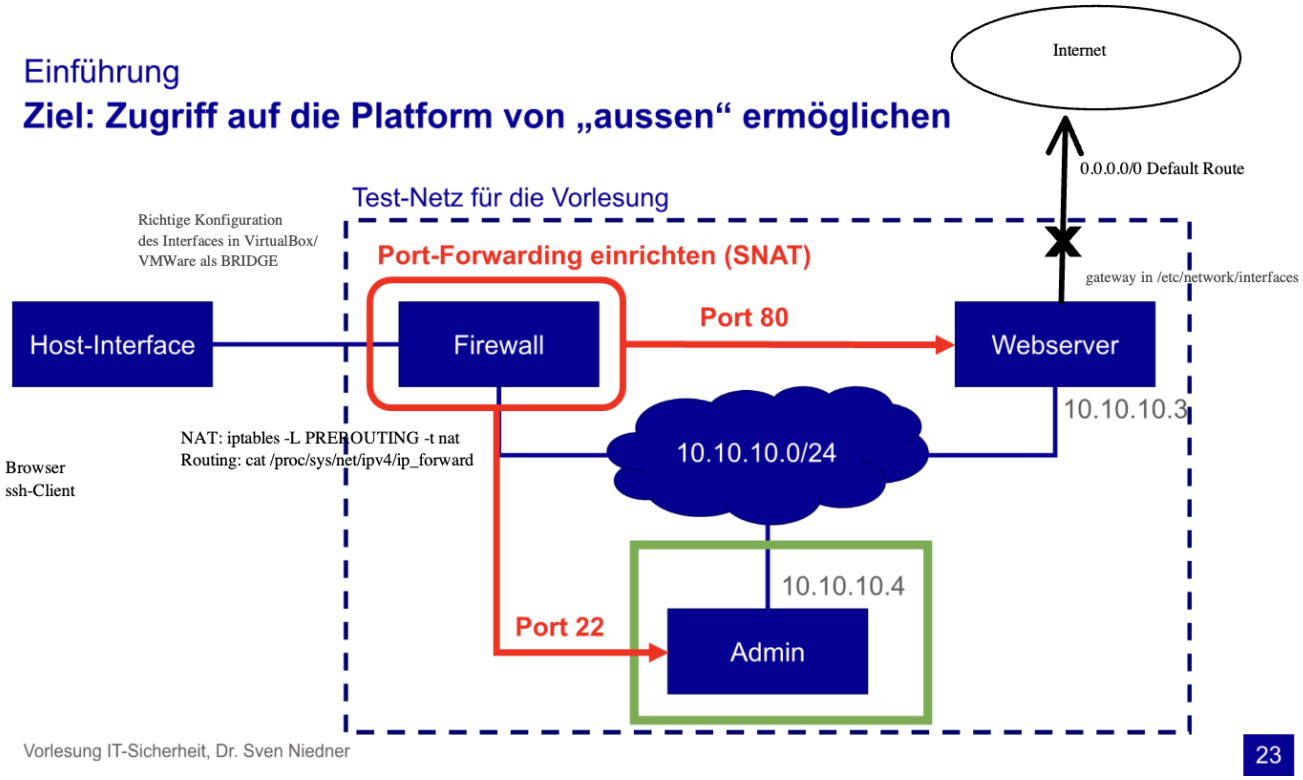
iptables Regeln anzeigen:

```
iptables -L PREROUTING -t nat
```

VMs Internet geben (auf Firewall):

```
iptables -t nat -A POSTROUTING -o <PUBLIC-INTERFACE-NAME> -j MASQUERADE
```

Schaubild



Härtung Firewall

Regelsätze

Einrichten einer Firewall-Regel

Aufgabe: Stellen Sie sicher, dass SSH-Zugriffe (Port 22) auf `firewall` und `webserver` nur vom `admin`-Rechner aus angenommen werden.

- Wie lautet die entsprechende Firewall-Regel?
- Recherchieren Sie die entsprechenden Optionen!
- Prüfen Sie das Ergebnis!
- Wie unterscheiden sich die Aktionen `DROP` und `REJECT`?

Firewall-Commands (auf Webserver und Firewall)

```
# Akzeptiert Verbindung von Adminserver
iptables -A INPUT -p tcp -s 10.10.10.4 --dport 22 -j ACCEPT

# Lässt keine erstmal keine Verbindung auf Port 22 zu
iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 22 -j DROP

# Für jeden weiteren Eintrag - Input auf 2 damit die 0.0.0.0/0 Regel an Pos 1 bleibt
iptables -I INPUT 2 -p tcp -s hier neue ip --dport 22 -j ACCEPT
```

Intrusion Detection System

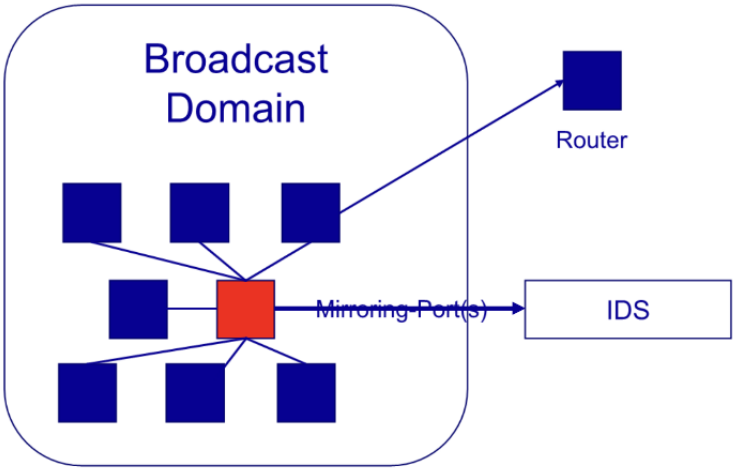
Netzwerksicherheit

Was ist ein IDS (Intrusion Detection System)?

- Grundgedanke: Erkennen von ungewollten Verbindungsversuchen
- Zugriffe auf „nicht für die Öffentlichkeit“ vorgesehen Dienste erkennen
  - Einstufung als Angriffsversuch
  - Kann bereits über die Logging-Funktion einer Firewall erfolgen
- Fortgeschrittene Funktion:
  - Auswertung der Datenverkehre auf verdächtige Muster (“Indicators of Compromise“)
  - Verdächtige DNS-Anfragen
  - Verdächtige Inhalte in http-Anfragen
- Herausforderung: Hohe Anzahl von Fehlalarmen
  - Alert Fatigue – man sieht den Wald vor lauter Bäumen nicht
- Beispielprodukte (Open Source): Zeek (ehem. Bro), Suricata

Netzwerksicherheit

Aufbau eines IDS für das Monitoring eines gesamten Netzes



Integration eines IDS ins Netz

- Über einen speziellen Port am Switch wird der gesamte Datenverkehr der Broadcast-Domain an das IDS ausgeleitet
- Datenverkehr ist nur in eine Richtung – ggfs. können Adern aus dem Netzwerkkabel entfernt werden
- Das IDS rekonstruiert den Datenverkehr und sucht nach verdächtigen Mustern
- Erfordert leistungsstarken Switch und leistungsstarken Rechner

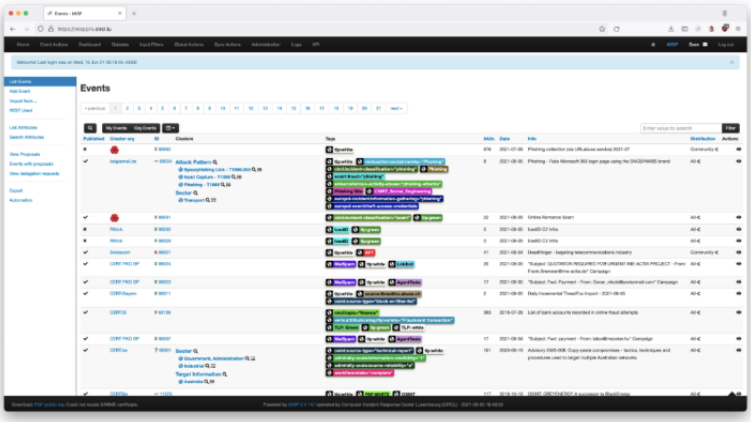
Aufbau eines Regelsatzes für ein IDS (Beispiel Suricata)

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET MALWARE Win32/TrojanDownloader.Agent.BXA CnC Activity";
flow:established,to_server; threshold:type limit, track by_src, seconds 600, count 1; content:"GET"; http_method;
content:"Xingapp|2f 35 2e 30 20 28|windowsxue|29|"; depth:24; isdataat:!1,relative; http_user_agent;
reference:md5,d4a8b93cb872a2817a1e7467ea449363; classtype:trojan-activity; sid:2033383; rev:1; metadata:affected_product
Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target Client_Endpoint, created_at 2021_07_21, deployment Perimeter,
former_category ADWARE_PUP, performance_impact Low, signature_severity Minor, updated_at 2021_07_21;)

alert tls $HOME_NET any -> $EXTERNAL_NET any (msg:"ET MALWARE Observed ZLoader CnC Domain in SNI"; flow:to_server,established;
tls_sni; content:"yuidskadjna.com"; isdataat:!1,relative; classtype:trojan-activity; sid:2033422; rev:1;
metadata:attack_target Client_Endpoint, created_at 2021_07_26, deployment Perimeter, former_category MALWARE,
signature_severity Major, tag SSL_TLS_SNI, updated_at 2021_07_26, mitre_tactic_id TA0011, mitre_tactic_name
Command_And_Control, mitre_technique_id T1573, mitre_technique_name Encrypted_Channel;)

alert dns $HOME_NET any -> any any (msg:"ET MALWARE Socelars Related Domain in DNS Lookup"; dns_query;
content:"www.cncode.pw"; nocase; depth:13; isdataat:!1,relative; reference:md5,f6c01214414fe2cedaa217c69ab093e1;
classtype:bad-unknown; sid:2033607; rev:1; metadata:attack_target Client_Endpoint, created_at 2021_07_28, deployment Perimeter,
former_category ADWARE_PUP, updated_at 2021_07_28, mitre_tactic_id TA0009, mitre_tactic_name Collection, mitre_technique_id
T1005, mitre_technique_name Data_from_local_system;)
```


Gelb: Indicator of Compromise

Quellen für IDS-Regeln und Indicators of Compromise

- Eigene Analyse auf Grundlage erkannter Angriffe und Vorfälle
- Bezug von Listen von IDS-Herstellern
- Offene Listen aus Open Source-Projekten
- Threat Sharing-Plattformen und Communities (bspw. MISP vom CERT in Luxemburg)
- Trend: Vernetzung der Sicherheits-Teams und gegenseitiger Austausch von „Indicators of Compromise“



Suricata

Netzwerksicherheit

Installation Suricata auf der Firewall

- Einrichten eines zusätzlichen, öffentlichen Netzwerkinterfaces vom Typ NAT (damit Zugriff auf das Internet möglich ist)
- Testen der Internet-Verbindung
- Installation Suricata:

```
apt install suricata suricata-update
```

- Editieren der Datei `/etc/suricata/suricata.yaml`, dort folgende Konfiguration eintragen:

```
address-groups:
      HOME_NET: "[10.10.10.0/24]"
```


ICMP-Alerts

Netzwerksicherheit

Test Suricata mit erster Regel

- Unsere erste Regel (Demo-Zwecke): Aufnahme einer Zeile in /etc/suricata/suricata.yml

```
rule-files:
    - ba-rm.rules
```

- Erstellen der Datei **/etc/suricata/rules/ba-rm.rule**s mit dem Inhalt:

```
alert icmp any any -> any any (msg: "ICMP Packet found";)
```

- Starten von Suricata mit:

```
suricata -i enp0s3 -i enp0s8 -D
```

- Suricata läuft jetzt im Hintergrund, beenden mit `killall suricata`
- Testen – Alarme finden sich unter `/var/log/suricata/fast.log`
- Ansehen mit `tail -f /var/log/suricata/fast.log` – Abbruch mit ctrl-C

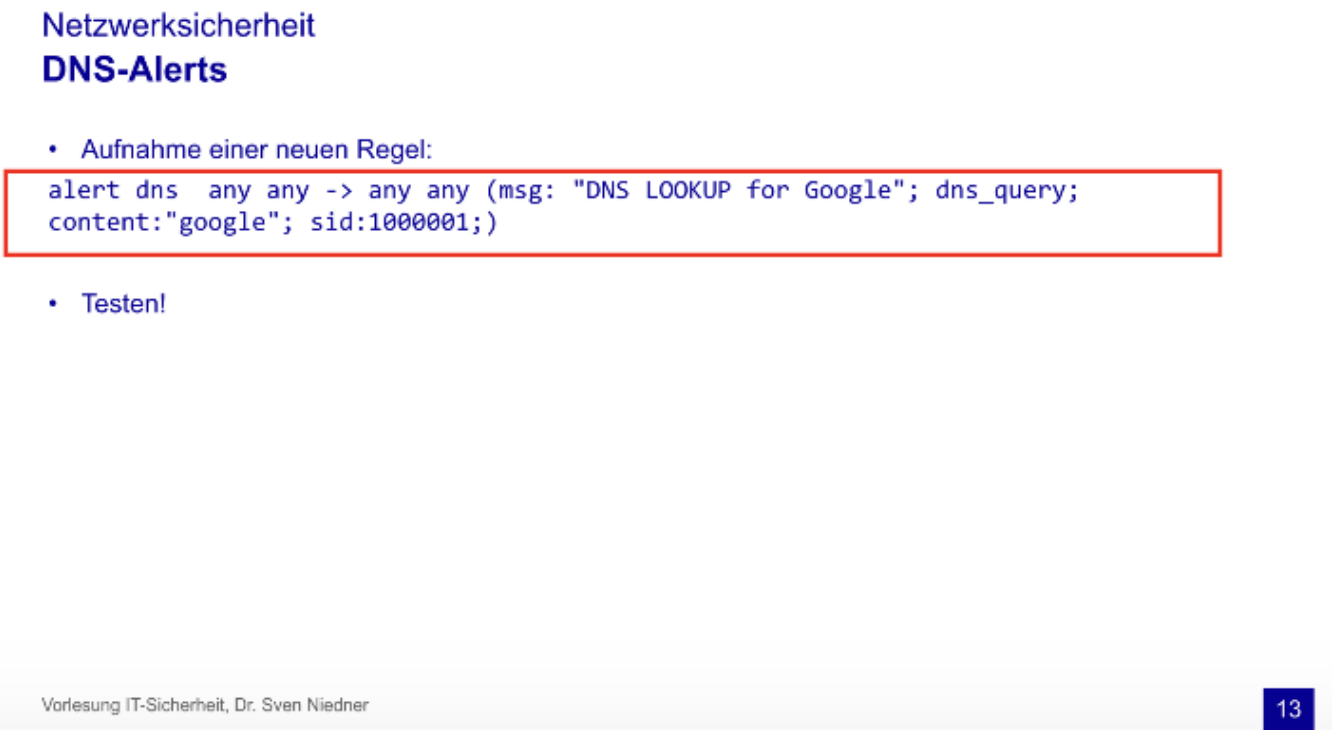

```
# Start von Suricata
# -D = Detatched


# suricata -i [INTERFACE1] -i [INTERFACE2] -D
```

```
suricata -i ens160 -i ens256 -D
```

Commands zum Beenden eines Prozesses

```
# Prozess anzeigen
ps aux | grep suricata

# Prozess killen
kill -9 [PID]
# oder
kill [PID]
```

DNS-Alerts



Diese Regel in Suricata überwacht den DNS-Verkehr (Domain Name System) und löst eine Warnung aus, wenn eine DNS-Anfrage den Begriff "google" enthält.

- **alert dns:** Dies gibt an, dass die Regel auf DNS-Verkehr angewendet wird.
- **dns any any -> any any:** Dies bedeutet, dass die Regel für jede DNS-Anfrage von jeder Quelle zu jedem Ziel gilt.
- **(msg: "DNS LOOKUP for Google";:** Dies ist die Nachricht, die in der Warnung angezeigt wird.
- **dns_query;:** Dies gibt an, dass die Regel nur auf DNS-Anfragen angewendet werden soll (keine Antworten).

- **content: "google";:** Dies ist die Bedingung, die erfüllt sein muss, damit die Warnung ausgelöst wird. Die DNS-Anfrage muss das Wort "google" enthalten.
- **sid:1000001;)** Dies ist eine eindeutige ID für die Regel.

Mit folgenden Befehlen kann man testen:

```
# Google pingen
ping google.com

# DNS-Auflösung
nslookup google.com
```

Ausgehende SSH-Verbindungen erkennen und alterten

```
alert tcp 10.10.10.0/24 any -> any 22 (msg:"Ausgehende SSH-Verbindung erkannt"; flow:established,to_client;
sid:1000002; rev:1;)
```

Suricata als Service in Debian machen

create user for **suricata**

```
useradd -r -s /usr/sbin/nologin suricata
```

change the **IFACE** at `/etc/default/suricata` and make it listen to our **infterface**

```
GNU nano 2.9.3                    /etc/default/suricata                    Modified

# Default config for Suricata

# set to yes to start the server in the init.d script
RUN=no

# Configuration file to load
SURCONF=/etc/suricata/suricata.yaml

# Listen mode: pcap, nfqueue or af-packet
# depending on this value, only one of the two following options
# will be used (af-packet uses neither).
# Please note that IPS mode is only available when using nfqueue
LISTENMODE=nfqueue

# Interface to listen on (for pcap mode)
IFACE=enp0s8

# Queue number to listen on (for nfqueue mode)
NFQUEUE=0

# Load Google TCMALLOC if libtcmalloc-minimal4 is installed
# This _might_ give you very very small performance gain....
TCMALLOC="YES"

# Pid file
PIDFILE=/var/run/suricata.pid
```

change the owner of **/var/log/suricata** using command below

```
chown -R suricata:suricata /var/log/suricata
```

edit **interfaces** at **/etc/suricata/suricata.yaml** and make it listen to our **infterface**

```
GNU nano 2.9.3                  /etc/suricata/suricata.yaml

  # This value is overridden by the SC_LOG_OP_FILTER env var.
  default-output-filter:

  # Define your logging outputs.  If none are defined, or they are all
  # disabled you will get the default: console output.
  outputs:
  - console:
      enabled: yes
      # type: json
  - file:
      enabled: yes
      level: info
      filename: suricata.log
      # type: json
  - syslog:
      enabled: no
      facility: local5
      format: "[%i] <%d> -- "
      # type: json


##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
  - interface: enp0s8
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    #  * cluster_flow: all packets of a given flow are sent to the same socket
    #  * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    #  * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #  socket. Requires at least Linux 3.14.
    #  * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    #  more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
    cluster-type: cluster_flow
    # In some fragmentation cases, the hash can not be computed. If "defrag" is set
    # to yes, the kernel will do the needed defragmentation before sending the packets.
    defrag: yes
```

start the S**uricata** and make sure it has been running well

```
systemctl start suricata
systemctl status suricata
```

Weiteres

Ausblick nächstes Semester

- Aufgabe: Gutes Ziel für einen Penetrationtest raussuchen - derjenige muss davon wissen
  - Ist prinzipiell illegal (ausnutzen von Sicherheitslücken) - mit dem Eigentümer des Systems muss es eine Vereinbarung geben
  - Gerät, Software, Docker-Image
  - Professionell gemanagte Dinge sind meist ziemlich sicher und schwierig etwas zu finden
  - Es werden Gruppen gebildet und wir gehen durch einen Pentesting Prozess inkl. Bürokratie
- Prüfungsleistung am Ende des Semesters ist eine Präsentation
  - Wissen aus diesem Semester ggf. mit abdecken
  - Penetrationtests

Kali Linux

- Gebaut für die Durchführung von Penetrationtests und Offensive Security
- Zahlreiche Tools vorhanden um Systeme anzugreifen
  - Nutzen gute und böse Personen (Schwachstellen aufdecken oder Missbrauch)
- In der Regel eine VM, Windows Subsystem for Linux hat mittlerweile auch eine Version

Tools

- aircrack-ng: WLAN Passwörter knacken
- Burp suite: Proxy zum Client und Server (Web) App Security - Anfragen manipulieren
- OWASP-ZAP: Web App Scanner
- Nmap, Wireshark: Netzwerk-Analyse
- Nessus: Schwachstellenscanner - gibt Gesamtbild über ein System ob etwas angreifbar ist (z.B. Datenbank)
- SQLmap: SQL Injection (vor allem Webseiten)
  - xkcd - Comic zur SQL Injection https://xkcd.com/327/

SQLMap

- **Erkennung von SQL-Injection-Schwachstellen:** Sqlmap kann automatisch Schwachstellen in Webanwendungen erkennen, indem es verschiedene Arten von SQL-Injection-Angriffen testet.
- **Ausnutzen von SQL-Injection-Schwachstellen:** Sobald eine Schwachstelle gefunden wurde, kann sqlmap diese ausnutzen, um auf die zugrunde liegende Datenbank zuzugreifen.
- **Datenbank-Interaktion:** Sqlmap bietet eine Vielzahl von Funktionen zur Interaktion mit der Datenbank, wie z.B. das Abrufen von Tabellen- und Spaltennamen, das Auslesen von Daten, das Ausführen von SQL-Befehlen und sogar das Hoch- und Herunterladen von Dateien.
- **Umfangreiche Datenbankunterstützung:** Sqlmap unterstützt eine Vielzahl von Datenbankmanagementsystemen, darunter MySQL, PostgreSQL, Microsoft SQL Server, Oracle und viele andere.

Webseite testen:

- Auf SQL-Injection-Schwachstellen scannen falls vorhanden und Namen der Datenbanken auflisten

```
sqlmap -u "http://192.168.108.129/" --dbs
```

- Scan mit Post-Daten
  - Befehl sendet POST-Daten an die URL `login.php`

```
sqlmap -u "http://192.168.108.129/login.php" --data="username=test&password=test" --dbs
```

- Scan mit Cookies
  - Befehl beinhaltet Cookies, die für den Zugriff auf die Seite erforderlich sein könnten. Ersetzen von `PHPSESSID=12345` durch die tatsächlichen Cookies

```
sqlmap -u "http://192.168.108.129/" --cookie="PHPSESSID=12345" --dbs
```

Log-Dateien ansehen:

- Liegen beim Webserver im /var/log/apache2/access.log
  - Zugriffe von SQLmap werden hier sichtbar

Härtung Suricata [SQL-Injection]

- SQL Injection über Kali ans Host-Netzwerk über die Firewall auf den Webserver
- Nutzung von SQLmap

Alerts für SQL-Injection

```
# Kritisch weil gibt schnellen Alarm wenn jemand irgendwas mit select schreibt
alert http any any -> any any (msg: "SQL-Injection erkannt! (content: Select)"; content:"Select"; nocase;
sid:1000004;)
# Leicht umgehbar weil userAgent geändert werden kann
alert http any any -> any any (msg: "SQLmap erkannt! (user_agent: sqlmap)"; http.user_agent;
content:"sqlmap"; nocase; sid:1000005;)
```

Weitere Möglichkeiten:

- Kommentare "--" im Content suchen (auch kritisch weil schneller Alarm)

Penetration Testing

Was ist Pentesting?

- Aufdecken von Sicherheitslücken
- Mögliche Angriffszenarien durchgehen
- Sicherheitslücken aufzeigen
- Dokumentation **aller** durchgeführten Schritte und Ergebnisse, jedes kleine bisschen
  - Fließt in die Präsentation ein
- "Ethical hacking"
- Eindringen mit Zustimmung
- Simulation eines Cyber-Angriffs ohne Schaden anzurichten
- Nutzung von Werkzeugen eines Angreifers
- Standards: OWASP Testing Guide, OSSTMM3
- Kunde möchte am Ende die Dokumentation der Ergebnisse: Was wurde getestet? Wie wurde es getestet? Was ist das Ergebnis?

Definition

- Verbesserung der IT-Sicherheit eines Systems
- Erkennung Schwachstellen
- "Nachweis" der Sicherheit durch eine 3. Partei (gelegentlich)
- Verbesserung interner Sicherheitsprozesse

Rechtliches

- Straftat Computerbetrug §263 StGB
- Ausspähen von Daten §202a StGB
- Abfangen von Daten §202b StGB
- Vorbereiten des Ausspähens oder Abfangens (Hacker-Paragraph) §202c StGB
- Datenhehlerei §202d StGB

Beauftragung

- Beauftragungsdatum hier: 10.01.2025
- Klärung Ziel, Zeitplan und Aufwand
- Festlegung des Ablaufs (Projektplan)
- Vertraulichkeitsvereinbarung (NDA)
- Rollen und Verantwortlichkeiten auf Tester und Kundenseite
- Umfang und Arbeitsaufwand

- Kommerzielle Aspekte: Kosten, Zahlungsmodalitäten, Haftung/Haftungsausschluss (im Projekt hier nicht)

Briefing

- Definition zu testende Systeme u. Netzwerkadressen (Scope)
- Klärung, in welchem Maß Zugang bestehen:
  - Internet?
  - Aus authentisierter Nutzer einer Anwendung?
  - Physikalischer Zugriff auf Netze im Haus?
- Sollen Maßnahmen mit zerstörerischen Auswirkungen durchgeführt werden?
- Was sind Kriterien und Vorgehen für Abbruch des Tests?
- DoS, DDoS eingeschlossen?
- Ausschluss bestimmter Techniken (z.B. Social Engineering, Gold Tickets, ...)
- Keine personenbezogenen Daten und Passwörter in die Dokumentation schreiben

Black Box Test

- Keine Informationen über das zu testende System
  - Strukturen, Schwachstellen, ... müssen von außen erkannt werden
- Vorteile
  - Perspektive eines externen Angreifers
  - Realistische Angrifssimulation
- Nachteile
  - Hoher Zeitaufwand durch aufwändige Analyse

White Box Test

- Vollständiger Zugriff auf Dokumentation, Quellcode, ... (Position eines gut informierten Insiders)
- Vorteile
  - Gezieltes Angreifen von Komponenten
  - Verständnis für den Gesamtnetzzusammenhang
- ...

Gray Box Test

- Zugriff zu ausgewählter Dokumentation bestimmter Systemkomponenten
- Vorteil
  - Gezielt Komponenten angreifen
  - Angriffswege können vorab identifiziert und geplant werden
  - Test kann auch bei unvollständiger Dokumentation durchgeführt werden
- Nachteil
  - Auftraggeber muss die bereitgestellte Dokumentation auswählen

Umfang des Pentests

- Uneingeschränkt
  - Alle Systeme können angegriffen werden
  - Ausnahmen können definiert werden für besonders kritische Systeme
- Eingeschränkt (meistens)
  - Nur bestimmte Systeme und Netze angreifen z.B. Alle Systeme in der DMZ
- Fokussiert (häufig)
  - Es wird nur eine bestimmte Anwendung oder Komponente getestet
  - z.B. Test einer neuen Webanwendung vor der Inbetriebnahme

Pflichten Auftraggeber und Auftragnehmer

- Auftraggeber
  - Bereitstellung Briefing-Informationen
  - Information von potentiell betroffenen Dritten (Cloud/Hoster)
  - Vermeidung unnötiger Schäden durch den Test
- Auftragnehmer
  - Geheimhaltung der erhaltenen Informationen und Schwachstellen
  - Dokumentation Vorgehensweise und Ergebnisse (Nachvollziehbarkeit)
  - Allgemeine Sorgfaltspflicht zur Vermeidung unnötiger Schäden

"Permission to attack"

- Angriffe gegen IT-Systeme sind normalerweise strafbar (§202a StGB)
- Bestätigung des Auftrags zum Angriff sichert den Pentester rechtlich ab
- Dokumentation sollte schriftlich erfolgen als sog. "permission to attack"

Durchführung

- Start-/Ende-Meldungen (z.B. Security Monitoring Team + Incident Management)
- Kontaktdaten für ggfs. Abbruch
- 5 Phasen
  - Vorbereitung
    - Tools installieren (Kali Linux)
    - WLAN aufsetzen in dem man die Geräte einbringen kann, ...
  - Reconnaissance
    - Überblick über die Zielsysteme bekommen
    - Analyse des Angriffsziels mit dem Ziel möglichst umfassende Informationen zu sammeln
    - Identifikation der vorhandenen Systeme die Teil des Angriffsziels sind
    - Nutzung öffentlicher Informationsquellen ("OSINT") im Internet
    - Verbindungsaufbau zu den Zielsystemen (Websiten-Aufrufe, Ports, Scans, ...)
  - Auswertung
    - Analyse erkannter Systeme und Schwachstellen

        - Auswahl von Angriffszielen, Dokumentation der Auswahl für den Abschlussbericht
        - Recherche von Angriffsmustern und Schadcode/Exploits
    - Angriffsversuche
        - Erfolgreicher Einbruch in die Zielsysteme
        - Aktive Einbruchsversuche mit ausgewählten Werkzeugen
        - Prüfung, in wie weit die Schwachstellen ausnutzbar sind
        - Prüfung, welche Konsequenzen die Ausnutzung der Schwachstellen für die Zielsysteme hat
        - Je nach Beauftragung kann auf die Durchführung des Angriffs verzichtet werden (Doku der Möglichkeit reicht dann aus)
    - Abschließende Analyse
        - Aufstellung der erkannten und angegriffenen Systeme
        - Darstellung identifizierter Schwachstellen
        - Ableitung damit verbundener Risiken
        - Darstellung nach (nachvollziehbare) Dokumentation erfolgreicher Angriffsversuche
        - Empfehlungen von Maßnahmen zur Behebung der Schwachstellen

Abschlussbericht

- Erklären was das Ziel ist
- Was sind die Briefing Informationen?
- Vorstellung der Ergebnisse des Pentests beim Kunden
- Vorgehen und eingesetzte Werkzeuge
- Identifizierte Systeme und Schwachstellen
- Bewertung der Schwachstellen
- Handlungsempfehlungen und Gegenmaßnahmen
- In Präsi: Kein Protokoll erzählen "das haben wir gemacht, und das ..." sondern eine gute Story erzählen

Tools - Kali Linux

- Optimiert für die Durchführung von Pentests
- Basiert auf Debian
- Läuft auch auf Windows WSL

Ausgewählte Tools:

- Aircrack-ng (WLAN Credentials)
- Burp suite (Application Security Scanner)
    - Technisch gesehen Proxy Server
    - MITM Angriff gegen TLS
    - Bildungsregeln für Token und Cookies
    - Manipulation und Veränderung von Anfragen
    - Einfügen von Strings, SQL-Injection

- OWAP-ZAP (Web Application Scanner)
  - Scan nach Standardproblemen (x-site-scripting)
- Nmap, Wireshark (Netzwerk-Analyse)
- Nessus (Schwachstellenscanner)
  - OpenVAS - Fork von der letzten OS-Nessus-Version aber OpenSource
- spiderfoot - Crawlen von Websites und extrahieren von interessanten Informationen
- ARP cache poisoning
  - Manipulation Auflöstung von ARP-Adressen auf IP-Adressen
  - Daten kommen auf Gerät des Angreifers an statt auf dem Standard-Gateway
  - Angreifer kann in geswitchten Netzen Daten mitlesen
  - Tool: arpspoof
  - Schutz davor: TLS
- fuzzer - z.B. API-Pentest Tool (viele Fehlereingaben senden)