# Sicherheit im Netzwerk

Wichtig: eigenes System auf Schwachstellen zu testen ist erlaubt, fremde Systeme darf man nicht scannen, das wird als Angriff gewertet und ist strafbar!!!

Security-Firmen machen sogenannte Pen-Tests (Penetration Tests), müssen dafür aber die Erlaubnis der Firma einholen, die sie testen.

Die website www.shodan.io liefert auch Infos zu Rechnern, die ungeschützt im Netz hängen, wenn der eigene dabei ist, ists nicht gut.

Die Linux-Distribution Kali Linux bringt Netzwerk Testing-Tools für jeden Zweck mit, etwa um Netzwerkverbindungen zu analysieren oder Systeme auf Schwachstellen zu prüfen. Für Nutzer von ARM-Rechnern, wie dem Raspberry Pi, gibt es abgespeckte Version.

Es geht aber auch viel mit den Standardnetzwerk Befehlen aus der Konsole, um Informationen über das Netzwerk und Rechner im Netzwerk zu bekommen

1. ipconfig /all:
   1. IP Adresse
   2. Gateway (da geht der gesamte Datenverkehr drüber)
   3. DHCP, DNS … Server (sind vielleicht auch für andere Aufgaben/Services zuständig)
2. tracert URL
   1. Hops zum Internet
   2. Adresse zur Firewall
3. netscan
   1. IP Adressen + Rechnername (ping –a auf alle Adressen im System) + Freigaben
   2. 255.255.0.0 = 65534 Adressen müssen durchsucht werden (0 und höchste Broadcast-Adresse fällt weg)

Probieren ob irgendwelche Server (🡪 angreifbar) über http (🡪 PW Klartext übertragen) erreichbar sind

WLAN ob irgendwo noch APs (Address Points) sind mit WEP oder WPA 🡪 Leicht zu knacken [Besser gesichert mit WPA2/CCMP/AES]

IoT Geräte, die bekannt für Schwachstellen (zum Teil erschreckend) sind (werden auch in shodan aufgelistet)

## Begriffe:

* SQL-Injection: wenn am Server zB eine query der Form   
  SELECT … WHERE user = ‘xyz‘   
  abgefragt wird und man als user nicgt nur xyz eingibt sondern  
  xyz ‘or 1=1‘wird die WHERE Klausel immer true sein und das SELECT etwas zurückgeben
* Cross-site scripting: Fremcode einschleußen, zB js Code über einen Link einschleußen, der den DOM am lokalen Client manipuliert zB indem man das Script als Argument dazuhängt URL?arg=“<script>böser js code</script>“  
  dann würde das script anstelle des Arguments im lokalen html code eingebaut werden

## Tools:

**Nmap (oder zenmap mit GUI) ->** analysiert das Netzwerk, offene Ports, vorhandene Dienste, Firewall

**Burp** -> testet webanwendungen zB js cross-site scripting (XSS) oder mysql injection mit zB or 1=1

**Wire Shark** (🡪 Sniffer) 🡪 Besser in der Analyse – Logger und Filter um spezielle Sachen zu verfolgen

**Cain & Abel** (🡪 Sniffer++) 🡪 „Man in the Middle“ ausprobieren, 2 [oder mehr] Adressen eingeben 🡪 Poisoning auswählen

## Cain und Abel

Sniffer zeigt Daten die übertragen werden

1

MITM

2

Cain & Abel MITM

* Configuration für http … Schlüsselwörter im HTML Stream z.B.: User, Password
  + 🡪 Gefundene Passwörter werden geloggt (zusätzlich ftp, pop, imap, ldap)
  + Besonders interessant ist ldap weil dort alle active directory Anfragen durchlaufen

## Andere Möglichkeiten

Brute Force Angrif auf die PW   
zB Lehrerkürzel sind bekannt. Geht mit einem einfachen VBS Script das eine LDAP Authentifizierung versucht geht auch mit LDAPs (siehe Anhang)  
Wörterbuchtabelle mit 1000 prominentesten Paswörtern: asdf, 1234, …  
Wenn Kennwort bekannt ist, kann man auch den e-mail account probieren, weil viele Benutzer das gleiche Kennwort für mehrere Anwendungen nutzen. Im Unternehmen gibt es oft einheitliche Namensbezeichnung für Benutzeraccount und email accounts zB [vorname.nachname@unternehmen.com](mailto:vorname.nachname@unternehmen.com)

Oder wenn man eine Passwortliste aus einem Datenbankserver hat, kann man Rainbow Tables testen, wo Klartextkennwörter mit zugehörigem Hashwert gespeichert sind)

man in the middle: Ein Angreifer setzt sich zwischen zwei Kommunikationspartner, durch Fälschen der ARP Einträge leitet er den Datenverkehr so um, dass alle Pakate an ihn gesendet werden und von ihm weitergeleitet werden.

mitmproxy installieren: dieser fängt http und https Verkehr ab und protokolliert ihn mit, geht auch für Handys

sslSplit macht das gleiche aber sogar auf der Transportebene und somit für alle Protokolle  
beide können auch falsche Zertifikate einschleußen.

## Allgemeine Empfehlungen:

https verwenden, verdächtige Zertifikate nicht ignorieren, Kaspersky Firewall ist zB stärker als die eingebaute Windows FW, und erkennt mitm Attacken öfter.

* PW regelmäßig ändern
* Sonderzeichen
* Nicht für alles das gleiche PW
* Aufpassen wo PW eingeben
* Virenscanner
* Als Administrator Anmeldeversuche verzögern zB 1 Sekunde – für den Benutzer nicht so schlimm, brute force dauert dann aber ewig. Nützt natürlich nichts, wenn das Passwort asdf ist.

## Vbs Script für LDAP Authentifizierungsversuch

Man versucht ein Active Directory user Object zu öffnen   
Beim Öffnungsversuch (unten rot) gibt man den Server, usernamen , Kennwort und Authentifizierungsmethode (secure oder nicht) mit  
wenns geht war das Kennwort richtig, sonst bekommt man einen Fehler beim Öffnen  
Schleife herumbauen und das Kennwort durchvariieren, zB aus einem Wörterbuch lesen  
on error goto 0

const ADS\_SECURE\_AUTHENTICATION = &h0001 '1

const ADS\_CHASE\_REFERRALS\_ALWAYS = &H60

dim objDSO

dim objUser

dim strPath

strServer="gandalf.htl-vil.local" ' geht schneller mit dem .htl-vil.local, weil dns nicht notwendig

strDomain="ou=Users, dc=htl-vil, dc=local"

strUser = "htl-vil\" & inputbox("Benutzer: ")

strPassword = inputbox("Kennwort:")

strPath = "LDAP://" & strServer & "/" & strDomain & ":636"

'der gesamte string in einem: LDAP://gandalf.htl-vil.local/ou=Users,dc=htl-vil,dc=local[:Port] wobei der optionale Port 636 (SSL) oder 389 ist

on error resume next

set objDSO = GetObject("LDAP:")

**set objUser = objDSO.OpenDSObject (strPath, strUser, strPassword, ADS\_SECURE\_AUTHENTICATION OR ADS\_CHASE\_REFERRALS\_ALWAYS)**

if err.number<>0 then

MsgBox "Fehler: " & err.description

else

MsgBox "Account ist gültig"

end if

## LDAP brute force mit js

Hier verwendet man die ldapjs Bibliothek:  
var ldap = require('ldapjs');  
Client erzeugen:  
client = ldap.createClient(Server und AD Struktur als parameter)  
und dann:  
client.bind mit verschiedenen usernamen und kennwörtern versuchen  
bind erfolgreich, wenn man User-Password Kombination passt

### Cain&Abel Screenshot

Im Network Hosts auflisten lasssen  
Im Sniffer die zwei Adressen auswählen, zwischen die man sich hängen will (zB Rechner und Gateway)  
Passwords anzeigen lassen die zB im http stream gefunden wurden  
im Configuration Dialog kann man einstellen nach welchen Schlüsselwörtern der Datenstrom durchsucht werden soll

