Sich selbst hacken: Scannen der eigenen Systeme

Mit Hackertools und -methoden kann man im eigenen Unternehmen Sicherheitslücken in der IT finden und schließen. Der erste Teil des neuen Tutorials „Sich selbst hacken“ zeigt, wie man herausfindet, was für Kriminelle von außen sichtbar und eventuell auch angreifbar ist.

Von Stephan Brandt

In der mehrteiligen Sich-selbst-hacken-Reihe für engagierte Admins, technisch versierte Sicherheitsverantwortliche und andere Verteidiger geht es darum, in den eigenen Systemen und Netzwerken Schwachstellen und zumeist offensichtliche Risiken wie veraltete Softwareversionen et cetera mit wenig Aufwand und kostenfreien Werkzeugen zu finden. Nach diesem Selbstaudit kann man die gefundenen Risiken bewerten und sich daran machen, sie samt ihren Ursachen zu beheben. Der Auftaktartikel stellt verschiedene Werkzeuge und ihre Nutzung vor, insbesondere für das Sammeln öffentlich verfügbarer Informationen, automatisierte Sicherheitsscans und das manuelle Verifizieren von Schwachstellen.

Vermeidbare Fehler finden

Laut der Studie „The State of Ransomware 2022“ von Sophos sind etwa 65 Prozent aller Hackingangriffe erfolgreich (siehe ix.de/zp11). Das durchschnittlich bezahlte Lösegeld bei einem kompromittierten Firmennetzwerk liegt bei ungefähr 800 000 US-Dollar. Wer in der Systemadministration arbeitet, hat vermutlich bereits Erfahrung mit diesem Thema gesammelt – hoffentlich ohne gravierenden Schaden. Eine der besten Methoden zur Sicherheitsüberprüfung der eigenen Systeme ist das Durchführen von Penetrationstests: Wenn es einem beauftragten Sicherheitstester gelingt, in das System oder Netzwerk einzudringen, schafft das auch ein böswilliger Hacker.

Ziel eines Ransomware-Angreifers ist es, in möglichst kurzer Zeit und mit möglichst wenig Aufwand große Geldsummen zu erpressen [1]. Dabei ist es nahezu unerheblich, bei welcher Firma es gelingt, ins Netzwerk einzudringen. Die zu erpressende Summe wird in jedem Fall beträchtlich sein. Kriminelle haben grundsätzlich zwei Möglichkeiten: Entweder sie greifen eine kleinere Anzahl von sehr lukrativen Opfern gezielt und mit großem Aufwand an oder sie versuchen möglichst viele Firmen mit überschaubarer Anstrengung, gegebenenfalls sogar automatisiert zu attackieren.

Die erste Option erfordert Erfahrung und viel Zeit, liefert jedoch die höchsten Lösegelder pro Angriff. Für die Massenangriffe benötigen Angreifer eine Reihe von Open-Source-Programmen, Rechenleistung, eine gute Internetanbindung – idealerweise mit einem Proxy in einem Land, in dem sie keine Strafverfolgung fürchten müssen – und etwas Geduld. Die profitabelsten Netzwerke kompromittieren sie auf diese Weise vermutlich nicht, aber die Erfolgschancen steigen, und damit die Anzahl der erfolgreichen Angriffe.

Ziel des Artikels ist es, sich vor der zweiten Art von Angriffen zu schützen. Einen Großteil der einfach auffindbaren Schwachstellen kann jeder selbst identifizieren und somit die Gefahr derjenigen Angriffe erheblich senken, die nur wenig Können erfordern. Die Herangehensweise ist denkbar einfach: Man scannt sein Netzwerk mit den gleichen Methoden, die ein Hacker verwendet, und behebt die entdeckten Probleme, bevor eine Ransomware-Gang sie sich zunutze macht.

iX-TRACT

- Systemverantwortliche haben verschiedene Möglichkeiten, sich selbst zu hacken und somit Schwachstellen in den eigenen Systemen und Netzwerken aufzudecken.
- Dazu stehen ihnen verschiedene Werkzeuge für Open Source Intelligence, automatisierte Sicherheitsscans und manuelles Verifizieren von Schwachstellen zur Verfügung. Neben kostenpflichtigen Diensten gibt es auch gute und leistungsfähige Tools für wenig oder kein Geld.
- Ziel ist es, leicht vermeidbare Fehler zu finden, ihre Gefährlichkeit zu überprüfen und die Schwachstellen samt ihren Ursachen zu beheben.

Der Werkzeugkasten für IT-Sicherheit

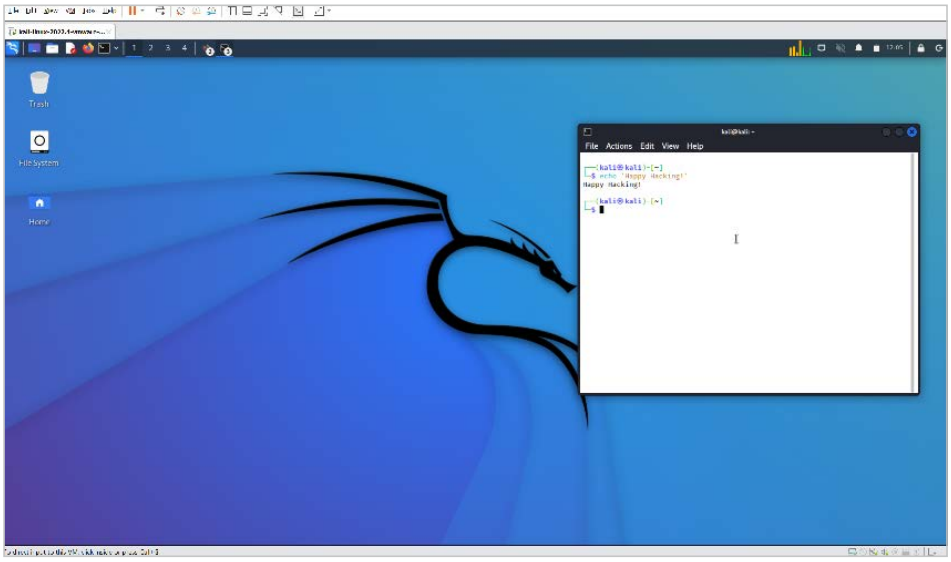
Für das Untersuchen von IT-Systemen auf Sicherheitslücken steht eine Reihe kostenloser Werkzeuge und Dienste zur Verfügung, zum Beispiel Kali Linux, eine spezialisierte Distribution für Sicherheitstests (siehe ix.de/zp11 und [2]). Hierbei handelt es sich um eine Linux-Variante, die auf Debian aufbaut und bereits eine große Anzahl an nützlichen

Open-Source-Programmen und Daten enthält. Die entsprechenden Pakete lassen sich auch einzeln manuell installieren und konfigurieren; es spart jedoch Zeit, dies nicht bei jedem Projekt aufs Neue zu tun. Sollte der Testverantwortliche nach einigen Sicherheitstests feststellen, dass man mit einem individuell konfigurierten System besser zurechtkommt, sollte man es aufsetzen. Für die meisten Anwendungsfälle genügt jedoch Kali in der Standardkonfiguration.

Kali Linux (Abbildung 1) kann wie jedes andere Betriebssystem direkt auf einem Rechner installiert werden. Komfortabler ist es jedoch, Kali in einer virtuellen Maschine (VM) zu betreiben. So kann man eine Basis-VM pflegen und für unterschiedliche Projekte oder Netzwerke, zum Beispiel für das Heim- und das Firmennetz, jeweils einen Klon der Basis-VM erstellen und an die Erfordernisse des speziellen Netzwerkes anpassen. Das stellt nicht nur eine organisatorische Trennung sicher, sondern grenzt auch die verschiedenen Netzwerke voneinander ab. Für den Fall, dass eine Test-VM in einem Netzwerk von Schadcode befallen wird, sind andere Netzsegmente davon nicht unmittelbar betroffen.

Für die Nutzung virtueller Maschinen ist eine Virtualisierungssoftware erforderlich. Die gängigsten kostenlosen sind der VMware Workstation Player und VirtualBox von Oracle. Für beide sind Lizenzen mit erweiterten Funktionen erhältlich. Vor Benutzung gilt es zu prüfen, inwieweit die Lizenzen mit dem konkreten Anwendungsfall kompatibel sind. Das Set-up der Basis-VM kann je nach Virtualisierungssoftware leicht variieren, die angewendeten Prinzipien sind jedoch identisch.

Auf der Webseite von Kali Linux (siehe ix.de/zp11) findet man DVD-Speicherabbilder (ISO) sowie fertige virtuelle Maschinen für VMware, VirtualBox und die in den Linux-Kernel integrierte Open-Source-Virtualisierungssoftware KVM/QEMU. Die einfachste Methode, Kali in Betrieb zu nehmen, ist, ein vorgebautes Image herunterzuladen und zu entpacken. Daraufhin lässt sich die VM direkt booten. Wenn man die VM mithilfe der ISO-Datei installiert, gestaltet sich die Konfiguration des Systems flexibler: Man kann zum Beispiel eine Festplattenverschlüsselung wählen und die Partitionierung bestimmen. Weil manche Tools Speicher fressen, weist man der Kali-Maschine mindestens 6 GByte RAM zu.



Ein bei Sicherheitsexperten beliebtes Werkzeug ist Kali Linux. In einer virtuellen Maschine betrieben lässt es sich flexibel für jeden benötigten Anwendungsfall anpassen und für regelmäßige Tests nutzen (Abb. 1).

Erst Passwort ändern, dann loslegen

Sobald der Tester die Kali-Maschine startet, kann er sich anmelden (Standard-Log-in für vorgebaute virtuelle Maschinen: kali:kali). Zuerst sollte er das Standardpasswort mit `passwd` zu einem sicheren Wert ändern. Auf keinen Fall sollte man fortfahren, ohne den Standardzugang zu schützen. Eine kompromittierte Basismaschine kann das gesamte Netzwerk gefährden. Außerdem sollte man für eine einfachere Handhabung persönliche Präferenzen konfigurieren. Falls etwa die Tastaturbelegung nicht passt, gibt man im Kali-Menü „Keyboard“ ein und korrigiert sie in der Registerkarte Layout. Vorsicht ist geboten, wenn man neue Passwörter mit einem ungewohnten Layout eingibt.

Ziel beim Konfigurieren der Netzwerkadapter ist es, die Kommunikation zwischen VM und Hostsystem zu unterbinden, der VM jedoch trotzdem Netzwerkzugriff zu erlauben. Am einfachsten und sichersten geht das, indem man einen USB-zu-LAN-Adapter an den Rechner anschließt und der VM zuweist. Hier ist sicherzustellen, dass die USB-Konfiguration der Virtualisierungssoftware mit dem Dongle kompatibel ist.

Außerdem sollte man einige Sicherheitsmaßnahmen beachten: Damit die VM nicht auf den Host zugreifen kann, entfernt man alle virtuellen Netzwerkadapter. Um nicht aus Versehen den Host in das zu testende Netzwerk einzubinden, sollte der USB-LAN-Adapter auf dem Hostsystem vollständig deaktiviert werden. Das geht in Windows über die Systemsteuerung im Menü „Netzwerk- und Freigabecenter“. Vorsicht: Windows-Updates können den Adapter reaktivieren.

Abschließend bringt man das neue System auf den aktuellen Stand und installiert eine Firewall. Kali nutzt das apt-Paketmanagement, entsprechend kann man das Update mit den Befehlen

```
sudo apt update && sudo ↵
                          apt upgrade
```

im Terminal ausführen. Mit

```
sudo apt install ufw
```

installiert man eine einfache Firewall und startet sie mit

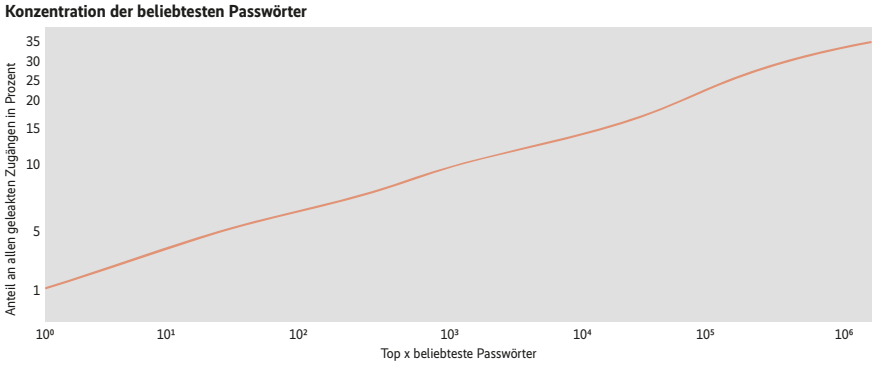
```
sudo ufw enable
```

OSINT: Was weiß das Internet über uns?

Viele der großen Angriffe der vergangenen Jahre hatten eines gemeinsam: Als initiales Einfallstor diente im weitesten Sinne Social Engineering. Dazu nutzen die Kriminellen häufig öffentlich verfügbare Informationen aus dem Internet. Diese Form der Informationsgewinnung wird als Open Source Intelligence Gathering, kurz OSINT, bezeichnet.

Für Social Engineering gibt es vor allem zwei niedrigschwellige Angriffsvektoren. Der einfachere von beiden besteht im Verwenden geleakter Passwörter, die über

Tutorialinhalt

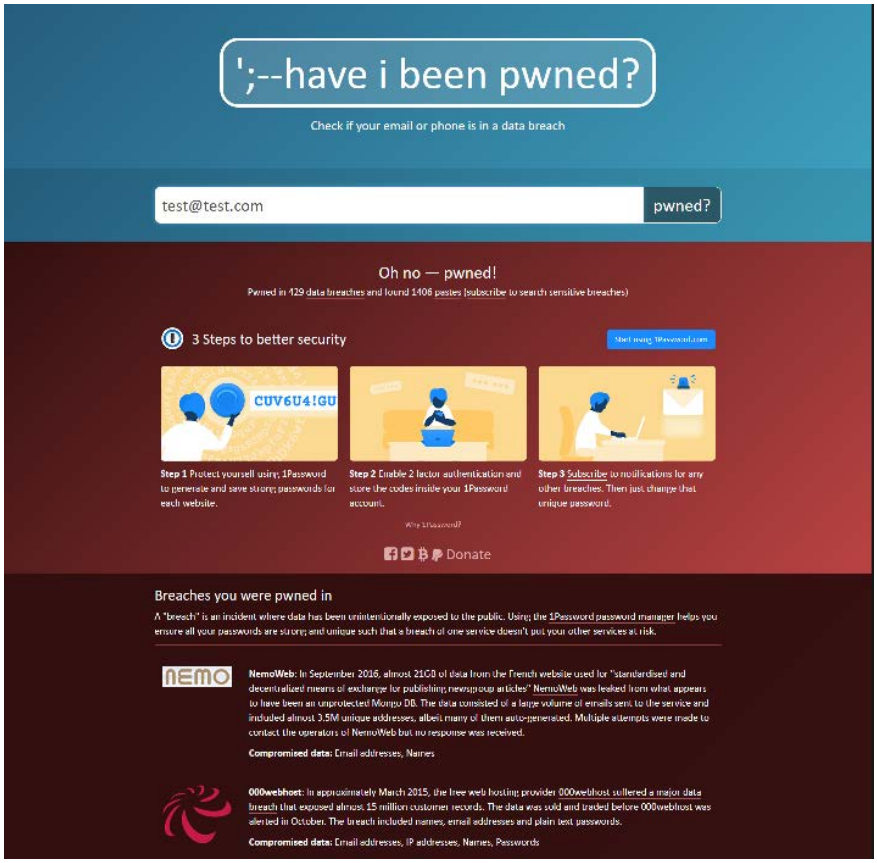


Anteil der Accounts, die eines der beliebtesten Passwörter verwenden (Abb. 2).

Onlinedatenbanken zugänglich sind. Etwas komplexer, aber ebenfalls leicht umsetzbar ist das massenhafte Versenden von Phishing-E-Mails.

Wie zahlreiche Reports in den vergangenen Jahren zeigen, sind die beliebtesten Passwörter sehr leicht zu erraten, beispielsweise „password“, „password1“ oder „123456“. Abbildung 2 wurde auf Basis der Hashdatenbank bei Have I Been Pwned (HIBP) erstellt (siehe ix.de/zp11). 35 Prozent der darin verzeichneten gehackten Konten nutzen eines der eine Million beliebtesten Passwörter (beide Achsen sind logarithmisch skaliert). Mit ausreichender Rechenleistung können diese innerhalb kürzester Zeit automatisch überprüft werden. Das Passwort auf Position 1000 000 wurde mehrere Hundert Mal entdeckt und hat damit höchstwahrscheinlich eine sehr einfache Struktur.

Wer also in einem Unternehmen mit einigen Hundert Mitarbeitern arbeitet, kann damit rechnen, dass einige von ihnen sehr einfach erratbare Passwörter benutzen. Sofern der Systemadministrator Zugriff auf die verwendeten Passwort-Hashes hat, kann und sollte er sie gegen die Hashes der Top 100 oder 1000 der beliebtesten Passwörter abgleichen und zum Passwortwechsel aufrufen [3].



Die Passwortabfrage von Have I Been Pwned kann entweder als Webanwendung oder direkt per eingebundener API genutzt werden (Abb. 3).

Geringer Aufwand – große Wirkung

Der zweite relevante Angriffsvektor, das Versenden von Phishing-E-Mails, ist vor allem deswegen so beliebt bei Kriminellen, weil diese Angriffsmethode skalierbar ist. Der Aufwand, eine E-Mail mit schadbringendem Anhang oder einem bösartigen Link zu erstellen, ist überschaubar. Gültige E-Mail-Adressen sind leicht über das Internet abrufbar und der Angriff kann problemlos auf mehrere Millionen Ziele ausgeweitet werden. In diesem Fall bedeutet selbst eine Erfolgsquote im Promillebereich eine Vielzahl kompromittierter Konten.

Falls ein Angreifer eine gezielte Attacke gegen eine Organisation ausführt, lässt sich die Erfolgsquote von Phishing-Mails auch deutlich in den zweistelligen Prozentbereich treiben. Dazu sucht der Angreifer auf öffentlich verfügbaren Kanälen (beispielsweise LinkedIn) nach einer Mitarbeiterliste und gegebenenfalls firmentypischen Umgangsformen, versucht eine E-Mail-Antwort der Firma zu provozieren, um an ein passendes Layout zu gelangen, und vieles mehr. Ist ihm ein interner Dienst bekannt, kann er dessen Oberfläche auf seiner bösartigen Seite nachbauen und so Zugangsdaten abgreifen. Diese können im Anschluss über das Darknet verkauft werden und landen häufig nach einiger Zeit auch in öffentlich zugänglichen Quellen.

Um also die Angriffsoberfläche für Phishing-E-Mails und Benutzerkonten mit öffentlich bekannten Passwörtern im eigenen Unternehmen einzuschätzen, sollte sich der Sicherheitstester ähnlicher Quellen bedienen wie potenzielle Angreifer. Es stehen verschiedene Dienste zur Verfügung, die E-Mail- und Passwortlecks im Auge behalten und strukturieren. Unter den gängigsten Tools finden sich die Seiten haveibeenpwned.com (HIBP) und dehashed.com.

Abfrage via Web oder per API

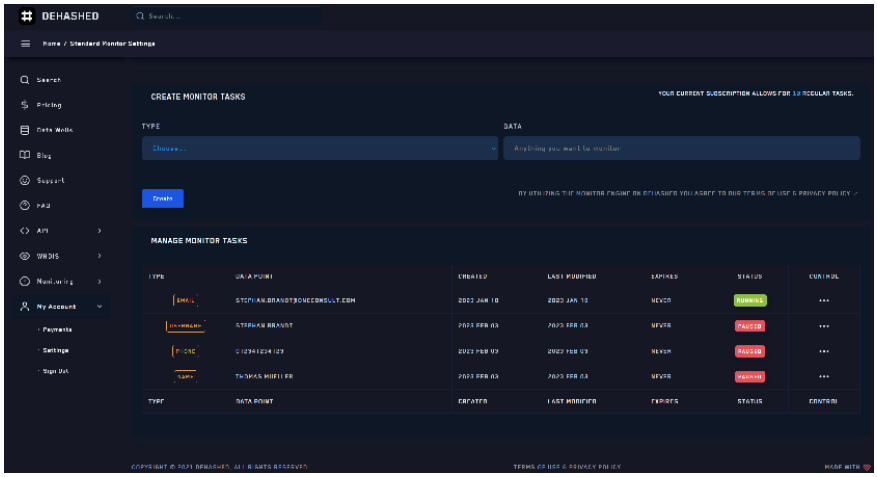
Eine Option ist, den Dienst Have I Been Pwned (HIBP) mittels Webformular abzufragen (Abbildung 3) oder die zugehörige kostenpflichtige API einzusetzen. Hierfür wird das eingegebene Passwort clientseitig in einen SHA1-Hash umgewandelt, übertragen werden lediglich die ersten fünf Zeichen des Hashs. Die API liefert anschließend die zweite Hälfte sämtlicher Hashes aus der Datenbank

zurück, die mit der entsprechenden Zeichenfolge beginnen. Abschließend wird wieder clientseitig überprüft, ob der Hash des eingegebenen Passwortes in der Liste steht.

Ein abgesichertes Verfahren

Das Backend der Applikation hat somit strukturelle Sicherheitsmaßnahmen implementiert, die eine Offenlegung der Passwörter verhindern, selbst wenn der Netzwerkverkehr mitgeschnitten wird. Für einen Test muss man jedoch nach wie vor der Implementierung im Frontend vertrauen. Diese könnte theoretisch das Passwort der testenden Person an unbefugte Stellen weiterleiten oder selbst Schwachstellen haben.

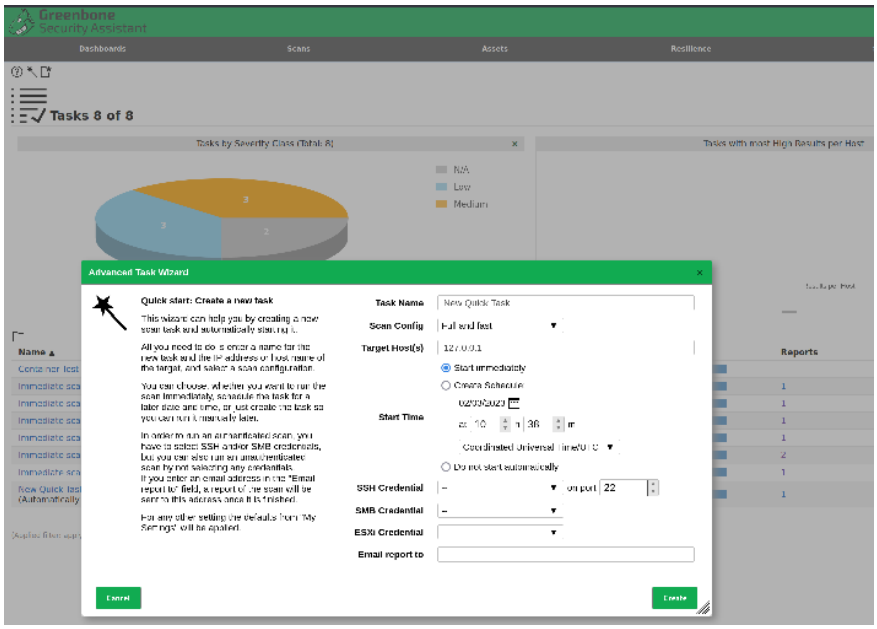
Alternativ lädt man die gesamte Passwort-Hash-Datenbank mit einer Größe zwischen 10 und 20 GByte als komprimiertes Archiv herunter – dazu nach unten scrollen (siehe ix.de/zp11). Nach dem Entpacken der heruntergeladenen Datei erhält man eine Textdatei mit Passwort-Hashes, sortiert nach der Anzahl der Leaks. In dieser Liste kann man mit dem grep-Befehl Hashes suchen. Aufgrund



Teuer, aber gut: die Monitoring-Dienste von dehashed.com. Allerdings werden die abgefragten Passwörter im Klartext ans Backend übertragen (Abb. 4).

der Größe der Datei ist sie jedoch ressourcenintensiv und unpraktisch. Sie eignet sich eher für einfache statistische Auswertungen.

Das letzte nützliche Feature auf HIBP ist ein Warnservice für verschiedene Kontentypen. Sowohl für einzelne Konten als auch ganze Domänen (siehe ix.de/zp11) kann man einen E-Mail-Alarm einrichten, der auslöst, sobald eine passende Adresse in einem Datenleck entdeckt wird. Aus Sicherheitsgründen muss man vor dem Einrichten einer „Domain Search“ nachweisen, dass man die Domäne kontrolliert und berechtigt ist, systematische Abfragen für diese durchzuführen. Für die Verifizierung stehen verschiedene Möglichkeiten zur Verfügung. Man kann eine Administrator-E-Mail-Adresse verwenden, einen Meta-Tag auf der eigenen Home Domain einfügen oder eine Datei oder einen



Bevor es losgeht, meldet sich der Testverantwortliche auf der OpenVAS-Management-Seite an und erstellt einen neuen Scan (Abb. 5).

TXT-Record mit einer entsprechenden UUID hochladen.

Permanentes Monitoring

Eine alternative Quelle ist dehashed.com (Abbildung 4). Im Gegensatz zu HIBP muss man sich für diesen Dienst registrieren und kann ohne kostenpflichtige Optionen nur wenige Details zu Datenlecks extrahieren. Analog zu HIBP enthalten die kostenlosen Funktionen einen Monitordienst, aber mit mehr Optionen: Man kann Benachrichtigungen für E-Mail-Adressen, Telefonnummern, Eigennamen und Nutzernamen einrichten. Zudem kann man parallel mehrere Methoden zur Benachrichtigung konfigurieren. Eine Monitoringaufgabe ist standardmäßig deaktiviert, man muss sie zuerst manuell aktivieren.

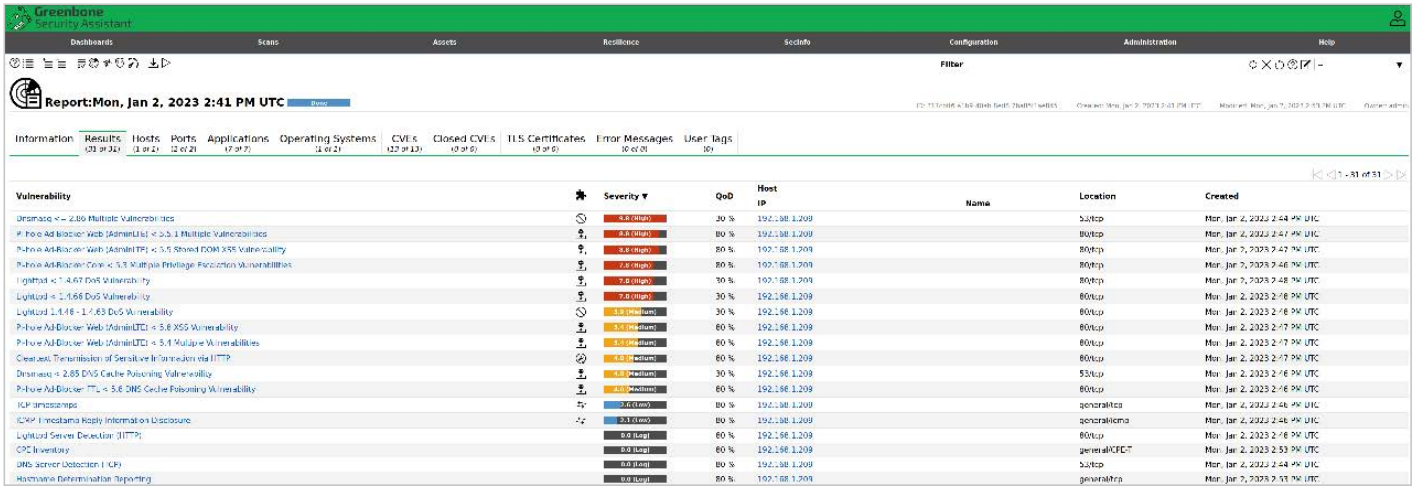
Für das Anzeigen aller Details in der Suchfunktion von dehashed.com benötigt man ein Standardabonnement (von etwa 5 US-Dollar wöchentlich bis 180 Dollar für ein Jahr). Mit diesem kann man in einer einheitlichen Suche gleichzeitig Nutzernamen, E-Mail-Adressen oder Domänen überprüfen. Zu jedem Eintrag zeigt die Seite zusätzliche Details an, etwa das geleakte Passwort oder die IP-Adresse. Die Suche nach aktiven Passwörtern ist bei DeHashed jedoch problematisch, da sie im Klartext als URL-Parameter an das Backend übermittelt werden. Der Anbieter selbst sowie ein Angreifer, der die Kommunikation mitlesen kann, könnte so Zugriff auf nicht bekannte Klartextpasswörter erhalten. Außerdem könnten diese Daten in Logs auf dem DeHashed-Server auftauchen oder im schlimmsten Fall über den Referer-Header des Browsers an externe Seiten geleakt werden.

Welcher der beiden Dienste geeignet ist, hängt vom Anwendungsfall ab. Wenn nur einzelne oder sehr wenige E-Mail-Adressen überprüft werden sollen, wird HIBP ausreichen. Auch das Prüfen von Passwörtern ist bei HIBP überzeugender umgesetzt, da dort keine Gefahr besteht, dass geheime und nicht bekannte Passwörter geleakt werden. Möchte man dagegen eine größere Firma und ihren Webauftritt umfassend prüfen und kann Ausgaben von mehreren Hundert Euro pro Jahr rechtfertigen, bietet DeHashed viel Flexibilität und eine Reihe zusätzlicher Funktionen.

Ein kommender Artikel der Selbst-Hacking-Reihe beschreibt, wie man durch öffentlich verfügbare Informationen IP-Adressen und Domänennamen von Servern und Webanwendungen der eigenen Organisation findet, die man als Schatten-IT womöglich gar nicht pflegt.

OpenVAS für den Einsatz vorbereiten

Nun geht es ans Überprüfen der eigenen Systeme. Zum einfachen Scannen eines Netzwerks bietet sich die Open-Source-Software OpenVAS an (siehe ix.de/zp11). Das OpenVAS-Projekt ist kostenlos und Teil der Schwachstellenmanagementprodukte von Greenbone Networks. OpenVAS ist verwandt mit der kommerziellen Software Nessus von Tenable Network Security. Es ist auf den neuesten Kali-VMs bereits vorinstalliert, kann andernfalls jedoch problemlos mit `apt` installiert werden. Für das Einrichten und



Ein Sachverhalt kann zu vielen Befunden führen, etwa bei diesen Scanergebnissen einer veralteten Pi-Hole-Version in einer Docker-Umgebung (Abb. 6).

Konfigurieren ist das apt-Paket gvm hilfreich, das man zusätzlich installiert.

Um die Systeme und Netzwerke jederzeit auf aktuelle Schwachstellen zu prüfen, muss man die zugehörige Datenbank regelmäßig aktualisieren. Die folgenden Befehle benötigen Administratorrechte, um Probleme mit Dateiberechtigungen zu vermeiden – idealerweise durch den root-Nutzer oder durch Voranstellen von `sudo` vor jedem Befehl.

Mit `gvm-setup` legt man automatisch alle notwendigen Nutzer an, das zugehörige Passwort wird angezeigt und die Datenbank synchronisiert. Dieser Prozess dauert beim ersten Durchlauf sehr lange, bis zu mehrere Stunden. Jede weitere Aktualisierung verändert jedoch nur die notwendigen Einträge und das Update läuft in wenigen Minuten durch. Bei vorgebauten Kali-VMs kann es beim gvm-Set-up zu Konflikten mit bereits vorhandenen Konfigurationen kommen. Um diese zu erkennen, kontrolliert man den Status der Konfiguration durch `gvm-check-setup`. Meist treten Konflikte zwischen existierenden OpenVAS-Nutzern und Zugriffsrechten auf bestimmte Ordner auf. Wenn bereits ein Nutzer für OpenVAS existiert und das Set-up deshalb keinen weiteren erstellen kann, muss man das Passwort dieses Nutzers ändern. Der entsprechende Befehl für den Admin-Nutzer lautet:

```
runuser -u _gvm -- gvmd --user=↵
  admin --new-password=<new_password>
```

Da der Scanner über den technischen Systemnutzer `_gvm` läuft, müssen einige Ordner via `chown` diesem Nutzer zugeordnet werden. Die betroffenen Ordner stehen in der Ausgabe von `gvm-check-setup`.

Wenn die Set-up-Prüfung keine weiteren Probleme findet, startet man den `gvmd`-Dienst und kann mit dem Scan des eigenen Netzwerks beginnen:

```
sudo systemctl start gvmd.service
```

Sollte der Dienst später im laufenden Betrieb Schwierigkeiten beim Start machen, deckt der Befehl `gvm-check-setup` einige Probleme automatisiert auf und repariert sie gegebenenfalls sogar.

Den Schwachstellenscan vorbereiten

Bevor man ein System oder Netzwerk auf Schwachstellen scannt, sollte man einige grundsätzliche Fragen klären. Welche Systeme oder Netze will man prüfen? Hat man die organisatorischen und technischen Berechtigungen, um die Rechner oder das Netzwerk zu scannen? Kommt die Kali-Testmaschine auf die entsprechenden Netze und hat keinen unnötigen Zugriff auf andere, sensible Bereiche? Ist das getestete System vollständig unter der Kontrolle desjenigen, der testet, oder ist ein externer Dienstleister zu benachrichtigen? Um einen störungsfreien Ablauf zu gewährleisten, sollte man sich überlegen, welche internen Stellen man informieren muss, weil sie etwa Firewalls oder Intrusion-Detection-Systeme anpassen müssen.

```
┌──(kali㉿kali)-[~]
└─$ host scanme.nmap.org
scanme.nmap.org has address 45.33.32.156
scanme.nmap.org has IPv6 address 2600:3c01::f03c:91ff:fe18:bb2f

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p- 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 05:49 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 300.29 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -p22,80,9929,31337 -sV -sC 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 05:57 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).

PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac00a01a82ffcc5599dc672b34976b75 (DSA)
|   2048 203d2d44622ab05a9db5b30514c2a6b2 (RSA)
|   256 9602bb5e57541c4e452f564c4a24b257 (ECDSA)
|_  256 33fa910fe0e17b1f6d05a2b0f1544156 (ED25519)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
|_http-favicon: Nmap Project
|_http-title: Go ahead and ScanMe!
|_http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.69 seconds
```

nmap-Scan einer Testseite – der Scanner untersucht auf den zu testenden Systemen alle Ports und ihren Zustand (Abb. 7).

Ein Scan kann außerdem ein System überlasten oder einen Ausfall verursachen. Für diesen Fall sollten Backups vorhanden sein. Ist ein zur Produktivumgebung identisches Testsystem vorhanden, das man risikofreier stattdessen scannen kann? Wer wäre von einer Störung oder einem Ausfall betroffen und wer könnte ihn zeitnah beheben? Zu welchen Tageszeiten laufen die Scans, wie bemerkt man Störungen und pausiert man den Scan dann kurzfristig?

Sobald alle betroffenen Stellen benachrichtigt und angemessene Vorkehrungen für etwaige Ausfälle getroffen sind, kann man mit den Scans beginnen. Dazu öffnet man in einem Browser die OpenVAS-Management-Seite unter https://localhost:9392 und meldet sich mit dem Nutzer aus dem Set-up-Prozess an. Die Warnung über das selbstsignierte Zertifikat kann man bestätigen. In der Aufgabenleiste ist nun die Registerkarte Scans auszuwählen und über das Zauberstabsymbol ein neuer Scan zu erstellen (Abbildung 5).

Zunächst kann man einige Scans auf unkritischen Testsystemen ausführen, um sich mit der Benutzeroberfläche vertraut zu machen und die Belastung des überprüften Rechners einzuschätzen. Man sollte dann mit einzelnen Systemen beginnen, bevor man große Netzwerke scannt.

Scannen ist kein Selbstzweck

Vor Beginn eines Scans und der konkreten Auswertung der Ergebnisse sollte das Ziel eines Schwachstellenscans klargestellt werden. Es ist keinesfalls möglich, sämtliche Verwundbarkeiten eines Systems automatisiert aufzudecken. Kein Vulnerability-Scanner kann Anspruch auf Vollständigkeit und Richtigkeit all seiner Ergebnisse erheben; seine Berichte enthalten mit hoher Wahrscheinlichkeit falsch positive Resultate. Das Ziel einer Schwachstellenprüfung sollte daher sein, leicht zu entdeckende Lücken wie veraltete Softwareversionen zu identifizieren. Für kritische, insbesondere extern erreichbare Dienste sollte man im Anschluss einen manuellen Penetrationstest durchführen (lassen). Je mehr „offensichtliche“ Schwachstellen zu diesem Zeitpunkt bereits behoben sind, desto eher können sich Tester bei ihrer manuellen Arbeit auf nicht triviale Lücken konzentrieren.

Die Ergebnisse eines Scans ruft man in der Weboberfläche über die Option „Scans/Reports“ ab. Es ist nicht erforderlich zu warten, bis die Prüfung abge-

Listing: Zielsysteme aus einer OpenVAS-Exportdatei extrahieren

```
import pandas as pd

results = pd.read_csv("<Report Datei>.csv")
slimmed_results = results.dropna(subset=["IP", "Port"])

targets = slimmed_results["IP"].astype(str) + ":" + slimmed_results["Port"].astype(int).astype(str)
targets = targets.drop_duplicates()

targets.to_csv('<Output-Datei>.csv', header=None, index=None)
```

schlossen ist, man kann die Ergebnisse live einsehen. Als Erstes gilt es zu prüfen, ob der Scan ordnungsgemäß gelaufen ist und ob die Testdauer für die Anzahl und Komplexität der überprüften Systeme plausibel erscheint. Wenn ein ganzes Netzwerk gescannt wurde, kontrolliert man in der Registerkarte Host, ob alle erwarteten Systeme entdeckt wurden und ob alte Systeme auftauchen, die Teil einer Schatten-IT sein könnten oder nicht mehr aktiv betreut werden.

Die Befundliste in OpenVAS ist standardmäßig auf eine Untermenge reduziert. Für einen ersten Überblick entfernt man alle auf die Ergebnisse angewandten Filter und macht eine erste Bestandsaufnahme.

Typischerweise führen einzelne Probleme zu vielen Befunden in einem Scan (Abbildung 6), besonders bei schwerwiegenderen Sicherheitslücken. Entdeckt OpenVAS beispielsweise eine veraltete Version 2.0 einer Software, zeigt der Scanner möglicherweise in mehreren einzelnen Befunden an, dass die Software anfällig ist, weil sie kleiner als Version 2.1 ist, kleiner als Version 3.0 und wegen Bugs in den Versionen von 1.8 bis 2.2. Die Bestandsaufnahme sollte sowohl vertikal erfolgen, wenn man viele Meldungen bei bestimmten Systemen oder in bestimmten Netzwerken findet, als auch horizontal: Die gleiche Sicherheitslücke tritt auf vielen verschiedenen Systemen auf.

Nach Priorität vorgehen

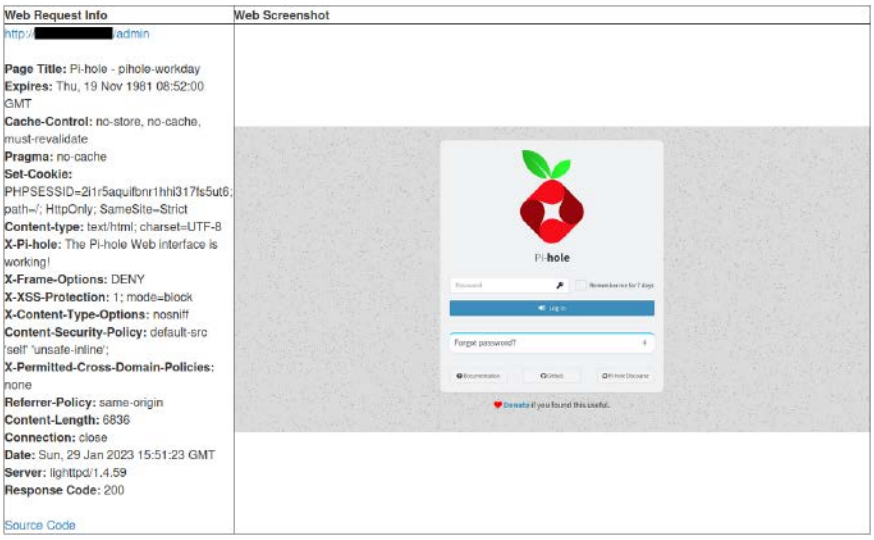
Hochkritische Funde wie Remote Code Execution auf einem öffentlich verfügbaren System sollten umgehend überprüft und Maßnahmen zum Beheben oder Eindämmen eingeleitet werden. Befunde, die keine unverzügliche Reaktion erfordern, können in Ruhe kategorisiert, priorisiert und in vorhandene Unternehmensprozesse einpflegt werden, beispielsweise in ein Ticketsystem oder Versionsmanagement. Zur Kenntnis nehmen sollte man auch die als „Low“ oder „Log“ eingestuften Schwachstellen. Hier finden sich zwar vermutlich keine direkt ausnutzbaren Lücken, aber möglicherweise eine Reihe Good Practices, deren Umsetzung zu einer deutlichen Härtung der Applikation beiträgt. Gängige Beispiele sind SSL/TLS-Chiffren, die nur noch für wenige Jahre empfohlen werden, oder nicht gesetzte sicherheitsrelevante HTTP-Header.

Für eine genaue oder eine teilautomatisierte Analyse der Testergebnisse bieten sich das Python-Paket python-gvm oder das zugehörige Kommandozeilenwerkzeug gvm-script an. Für das Python-Paket stellt Greenbone auf GitHub eine große Anzahl an Skripten zur Verfügung, die man auch über die Kommandozeile ausführen kann. Dabei hilft die umfangreiche Dokumentation (siehe ix.de/zp11). Außerdem lassen sich Berichte über die Weboberfläche exportieren. Es erscheint ein Dialogfeld, in dem man zwischen mehreren Datentypen wählen kann, unter anderem XML und CSV. Diese Dateien kann man mit eigenen Skripten weiterverarbeiten.

Scannt man Geräte und Netzwerke fortlaufend, kann man die Entwicklung der Netzwerksicherheit beurteilen. Beim Anlegen eines Scans lässt sich ein Zeitplan definieren, nach dem er wiederholt wird. OpenVAS erstellt im Laufe der Zeit automatisch eine Reihe von Grafiken zu den getesteten Netzwerken und Systemen. Nachhaltig steigern lässt sich die Sicherheit des eigenen Netzwerkes nur dann, wenn man den Schwachstellenscan in einen iterativen Prozess aus Identifikation (Scan), Priorisierung und Maßnahmenplanung, Problembehebung (zum Beispiel Updates) und Überprüfung (Rescan) einbindet.

Scanergebnisse verifizieren: Was kann man glauben?

Die Ergebnisse eines Scans kann man für Systemverantwortliche in drei grobe Kategorien einteilen. Bestimmte Befunde lassen sich vermutlich auf Anhieb als False Positives ausschließen. Beispielsweise ist es wenig plausibel, wenn eine Warnung zu einem veralteten Windows-Betriebssystem für eine Linux-Maschine gemeldet wird. Auf der anderen Seite sind vielleicht bestimmte Probleme bereits bekannt und in Bearbeitung, oder sie sind aus geschäftskritischen Gründen noch nicht behoben worden. Beide Kategorien lassen sich über rein organisatorische Kanäle abwickeln, da alle technischen Fragen bereits beantwortet sind.



Ergebnis eines EyeWitness-Reports, der mit dem zugehörigen Screenshot ausgegeben wird (Abb. 8).

Befunde der dritten Kategorie, die plausibel wirken, aber nicht bestätigt wurden, sollte man händisch verifizieren. Sofern der Testende Zugriff auf das betroffene System hat, kann er Punkte wie Versionsnummern leicht feststellen. Ist das nicht möglich, sollte er den Befund über einen zweiten Kanal bestätigen. Für einen Großteil der Anforderungen eignet sich der Open-Source-Portscanner nmap (Abbildung 7).

nmap ist bereits auf Kali vorinstalliert und kann ohne weitere Konfiguration eingesetzt werden. Die wichtigsten Funktionen ermöglichen das Scannen von Ports und deren Zustand, die Identifikation von Diensten und das Überprüfen der entdeckten Services durch nmap-Skripte.

Alle offenen TCP-Ports auf einem System findet der Befehl

```
sudo nmap -sS -p- <IP Adresse des ↵
                          Zielsystems>
```

wobei die Option `-sS` den Scantyp als SYN-Scan festlegt und `-p-` alle zur Verfügung stehenden TCP-Ports scannt. Da diese Art von Portscan Teile des Netzwerkstacks des Betriebssystems umgeht, sind Root-Rechte erforderlich.

Portscans im Detail

Möchte man bestimmte Ports spezifizieren, kann man diese kommasepariert angeben, beispielsweise `-p80,443`. Ohne Parameter prüft nmap standardmäßig die tausend gängigsten Ports eines Systems. Der Portparameter ist vor allem dann hilfreich, wenn man eine große Anzahl an Systemen in begrenzter Zeit scannen möchte. Neben dem TCP-SYN-Scan, der die ersten zwei Schritte eines TCP-Handschlags (SYN und ACK) ausführt, gibt es eine Reihe weiterer Optionen. Die wichtigste davon ist der UDP-Scan (Option `-sU`). Da UDP-Ports im Gegensatz zu TCP-Ports kein Antwortpaket zurückschicken, versendet nmap für diesen Scan lediglich leere UDP-Pakete und wartet auf eine Antwort. Dadurch sind solche Scans sehr langwierig. Um Zeit zu sparen, lassen sich SYN- und UDP-Scans in einem Befehl kombinieren:

```
sudo nmap -sSU -p- <IP Adresse des ↵
                          Zielsystems>
```

Mit den oben genannten Befehlen liefert nmap eine Liste der auf dem System entdeckten Ports zurück, zusammen mit dem Zustand des Ports und dem Dienst, den es aufgrund der Portnummer dahinter vermutet. nmap unterscheidet hauptsächlich zwischen drei verschiedenen Zuständen, die ein Port einnehmen kann: „open“ bedeutet, dass der Dienst auf eine nmap-Anfrage geantwortet hat, „closed“, dass der Port erreichbar war, die Verbindung jedoch abgelehnt wurde, und „filtered“ beschreibt Ports, die auf keine Anfrage von nmap reagieren. Letzteres kann beispielsweise eine Firewall verursachen, die sämtliche Pakete für einen bestimmten Port fallen lässt.

Nachdem alle Dienste aufgelistet wurden („enumeriert“), kann man sie über Banneranalyse und nmap-Skripte weiter untersuchen:

```
sudo nmap -p<identifizierte Ports ↵
    als Komma-separierte Liste> -sV ↵
                        -sC <IP Adresse>
```

Der Parameter `-sV` liefert das Banner des Dienstes zurück, wodurch sich die verwendete Software und gegebenenfalls die zugehörige Version identifizieren lässt. Die Option `-sC` führt gängige nmap-Skripte für den vermuteten Dienst aus. Dadurch kann man bestimmte Schwachstellen und Fehlkonfigurationen finden.

Mit diesen Optionen begibt man sich manuell auf die Suche nach offenen Ports und potenziell verwundbaren Diensten. Man sollte ruhig damit an den eigenen Systemen experimentieren, um ein Gefühl für den Zeitaufwand und die Zuverlässigkeit der Ergebnisse zu bekommen. Sucht der Testverantwortliche Schwachstellen in einer Produktivumgebung, sollte die Arbeit im Nachhinein nachvollziehbar und sollten sämtliche Ergebnisse dokumentiert sein. Hierfür bietet nmap die Option `-oX <filename>`, um eine XML-Datei zu erstellen. Für weitere Dateiformate lohnt ein Blick in die auch sonst lesenswerte Onlinedokumentation (siehe ix.de/zp11).

Unbekanntes erkennen: automatisierte Screenshots

Hat man ein Netzwerk entweder mit OpenVAS oder mit nmap gescannt und dabei unbekannte Systeme entdeckt, gilt es, diese möglichst schnell zu identifizieren und einzuschätzen. Dafür wurde EyeWitness entwickelt (siehe ix.de/zp11). Das Werkzeug öffnet die Adressen der entdeckten Systeme und zugehörige Ports in einem Browser und erstellt einen Screenshot. Zudem sammelt es Informationen über das System, zum Beispiel HTTP-Header oder die Serverversion, und nimmt sie in den Bericht auf. Für bestimmte Dienste probiert es automatisch bekannte Standardnutzer und -passwörter aus.

Eine besonders gelungene Funktion ist, dass das Programm die Systeme direkt aus einem nmap-Scan übernehmen kann. Hierfür ist lediglich die von nmap exportierte XML-Datei zu übergeben:

```
eyewitness -x <nmap-XML-Datei>
```

Für OpenVAS-Scans ist das leider nicht so einfach möglich. EyeWitness kann jedoch auch mit einer einfachen Liste aus Adressen arbeiten. Diese kann man sich mithilfe von CSV-Exporten aus der OpenVAS-Weboberfläche und dem kleinen Skript selbst erzeugen. Aufgrund der Struktur des CSV-Exports können Standardwerkzeuge in der Kommandozeile Probleme verursachen. Der Python-Code im Listing kann die IP-Adressen und Ports aus einer Exportdatei zu einem Dokument aus Zielsystemen kombinieren. Diese Datei übergibt man mit der Option `-f <Dateiname>` an EyeWitness.

Den von EyeWitness erzeugten Bericht kann man direkt im Browser öffnen und untersuchen. Ein Beispiel für die Ergebnisse ist in Abbildung 8 dargestellt.

Fazit

Sowohl Systemverantwortliche als auch Hobby-ITler können die eigenen Systeme ohne größeren Aufwand an Zeit und Geld scannen. Dadurch erkennen sie offensichtliche Probleme im Netzwerk frühzeitig und können veraltete Software aktualisieren oder auch kritische von außen zugängliche Ports schließen. Das schützt bereits vor vielen einfachen Angriffen.

Der nächste Artikel der Reihe zeigt, wie man Webanwendungen genauer unter die Lupe nimmt. (ur@ix.de)

Quellen

[1] Tam Hanna; Ransomware-Rundblick; iX 10/2022, S. 150

[2] Frank Neugebauer; Mit Dampf; Kali Linux 2021.1; iX 5/2021, S. 58

[3] Sandro Affentranger; Schwierige Wahl; Passwortsicherheit (nicht nur) im Active Directory; iX 1/2022, S. 116

[4] Alle zitierten Werkzeuge, Dokumentationen und Websites sind über ix.de/zp11 zu finden.

STEPHAN BRANDT

ist Penetrationstester bei der Oneconsult Deutschland AG. Neben den Tests beschäftigt er sich mit der automatisierten Auswertung und Dokumentation von Scanergebnissen.