Linux-Sicherheitslücken und Passwortsicherheit

Lösen Sie die nachfolgenden Aufgaben und bereiten Sie diese bis zum nächsten Lehrveranstaltungstermin vor.

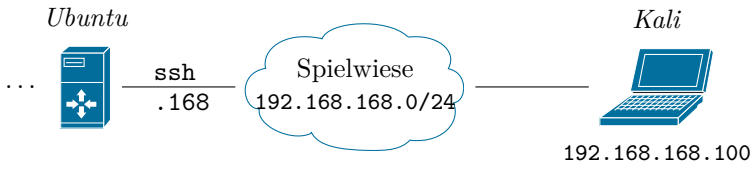
01.

a) Installieren Sie *PolicyKit* nach der zur Verfügung gestellten Anleitung auf dem *Ubuntu*-Server zur Privilegienverwaltung.

b) Konfigurieren Sie den SSH-Server auf dem *Ubuntu*-Server derart, dass die IP der in der Grafik unten angegebenen entspricht. Folgen Sie den Hinweisen zur Einrichtung.

c) Öffnen Sie mit dem Befehl `ssh <Benutzer>@ssh.<Domänenname>` (z.B. `ssh wartung@ssh.spielwiese-unterweger.tld`) als Benutzer `bruce` von der *Kali*-VM aus eine SSH-Sitzung auf dem *Ubuntu*-Server. Lassen Sie sich mit dem Befehl `uname -a` die genaue Betriebssystemversion des *Ubuntu*-Servers ausgeben und diskutieren Sie die Sicherheitsimplikationen dieser Information.

d) Versuchen Sie in der SSH-Sitzung mit dem Befehl `cat` (der einen Dateinamen als Parameter verlangt), die beiden Dateien `/etc/passwd` und `/etc/shadow` auszugeben und beenden Sie die Sitzung mit `exit`. Kopieren Sie anschließend alle Dateien, auf die zugegriffen werden konnte, mit dem Befehl `scp ssh.<Domänenname>:<Dateipfad> <Zielpfad>` (z.B. `scp ssh.spielwiese-unterweger.tld:/etc/shadow /pwds.txt`) unter Angabe eines Benutzers (wie bei `ssh`) auf die *Kali*-VM und überprüfen Sie den Inhalt der kopierten Datei(en). Diskutieren Sie die Sicherheitsimplikationen der gewonnenen Informationen und die Dateizugriffsrechte.



02.

a) Starten Sie *Metasploit* auf der *Kali*-VM und loggen Sie sich damit erneut als Benutzer `bruce` auf dem *Ubuntu*-Server ein. Verwenden Sie dazu das Modul `auxiliary/scanner/ssh/ssh_login` und überprüfen Sie die Verbindung, indem Sie das Kommando `pwd` an alle aktiven Sitzungen schicken, um sich das jeweils aktuelle Verzeichnis (*print working directory*, kurz `pwd`) auf dem Server ausgeben lassen. Folgen Sie den Hinweisen zur Verwendung.

b) Suchen Sie anhand der bisher gesammelten Informationen zur Betriebssystemversion des *Ubuntu*-Servers ein Exploit-Modul aus `exploit/linux/local/` (**außer** `pkexec`), dessen Veröffentlichungsdatum jünger ist als das der letzten Betriebssystemaktualisierung. Führen Sie den Exploit durch und erläutern Sie, ob bzw. warum dadurch (k)eine *meterpreter*-Sitzung geöffnet werden konnte. Folgen Sie den Hinweisen zur Verwendung.

c) Führen Sie einen erneuten Exploitversuch mit dem Modul `exploit/linux/local/pkexec` durch. Wiederholen Sie diesen bis zum Erfolg und wechseln Sie anschließend in die geöffnete *meterpreter*-Sitzung. Stellen Sie dort mit dem Befehl `getuid` sicher, dass Sie nun Superuser-Rechte besitzen und laden Sie mit dem Befehl `download` alle Passwortdateien (vgl. vorherige Beispiele) herunter. Diskutieren Sie die Sicherheitsimplikationen der gewonnenen Informationen sowie der durchgeführten Aktionen. Folgen Sie den Hinweisen zur Verwendung.

d) Verwenden Sie das Modul `post/linux/gather/hashdump` mit der geöffneten *meterpreter*-Sitzung, um eine strukturierte Sammlung von Passwörtern durchzuführen. Lassen Sie sich nach dem Schließen der Sitzung mit dem Befehl `loot` anzeigen, welche Informationen gesammelt wurden und überprüfen Sie den Dateiinhalt des *Unshadowed Password File*. Diskutieren Sie die Sicherheitsimplikationen der gewonnenen Informationen und folgen Sie den Hinweisen zu Verwendung.

03.

a) Verwenden Sie *John the Ripper*, um aus der vom *Ubuntu*-Server ermittelten *Unshadowed Password File* die Klartextpasswörter aller Benutzer zu ermitteln. Diskutieren Sie die Passwortsicherheit der gewählten Passwörter und die damit einhergehenden Sicherheitsimplikationen. Folgen Sie den Hinweisen zur Verwendung.

b) Lassen Sie die drei Passwörter `password` (im Standardmodus), `0815` und `Kufstein` (jeweils im Brute-Force-Modus) als Linuxpasswörter (d.h. als SHA-512-Hashes) von *John the Ripper* im Klartext rekonstruieren und halten Sie fest, wie lange die Ermittlung jeweils dauert. Diskutieren Sie die Passwortsicherheit dieser Passwörter und die damit einhergehenden Sicherheitsimplikationen. Folgen Sie den Hinweisen zur Verwendung.

c) Erstellen Sie anhand Ihrer bisherigen Erkenntnisse ein möglichst sicheres Passwort und versuchen Sie, dieses als Linuxpasswort mit *John the Ripper* im Standard- und im Brute-Force-Modus (jeweils maximal 15 Minuten) im Klartext zu rekonstruieren. Halten Sie fest, wie lange die Ermittlung jeweils dauert. Diskutieren Sie die Passwortsicherheit und die damit einhergehenden Sicherheitsimplikationen. Kann Ihr gewähltes Passwort auch nach geraumer Zeit nicht ermittelt werden, setzen Sie dieses für den Benutzer `bruce` auf dem *Ubuntu*-Server als Benutzerpasswort. Verwenden Sie dazu den Befehl `passwd` in einer (neuen) SSH-Sitzung.

Hinweis zur Einrichtung des SSH-Servers auf einem *Ubuntu*-Server

Stellen Sie sicher, dass Sie eingeloggt sind und führen Sie alle nachfolgenden Befehle als Superuser aus. Öffnen Sie die Konfigurationsdatei des SSH-Servers mit dem Editor `nano`, indem Sie

```
nano /etc/ssh/sshd_config
```

eingeben.
Entfernen Sie die Raute (`#`) vor der Zeile für die Option `ListenAddress` und ersetzen Sie `0.0.0.0` durch die IP-Adresse, auf der der SSH-Server lauschen soll.
Nach der Konfiguration dieser Änderung muss der SSH-Server neu gestartet werden:

```
/etc/init.d/ssh restart
```

Abschließend kann mit dem Befehl

```
netstat -4at
```

überprüft werden, ob nur ein SSH-Port auf der gewünschten IP-Adresse offen ist und keine weiteren IP-Adressen bedient werden.

Hinweis zur Verwendung von *Metasploit* – 1. Initialisierung

Vor dem Start von *Metasploit* muss der *PostgreSQL*-Dienst gestartet werden, der benötigte Datenbankfunktionalitäten zur Verfügung stellt:

```
service postgresql start
```

Beim allerersten Start muss außerdem **einmalig** die Datenbank von *Metasploit* initialisiert werden:

```
msfdb init
```

Nach diesen Schritten kann die Konsolenversion des *Metasploit Framework* (kurz MSF) mit dem Befehl

```
msfconsole
```

gestartet werden.
Nach dem Start der Konsole kann mit dem Befehl `db_nmap` (Verwendung wie `nmap`) eine Reconnaissance im Netzwerk durchgeführt werden. *Metasploit* speichert die Ergebnisse dabei automatisch in der Datenbank. Nach dem Scan können die gefundenen Rechner mit dem Befehl

```
hosts
```

angezeigt werden. Um sie automatisch als mögliche Ziele für Verbindungen, Angriffe u.ä. zu berücksichtigen, können sie der Liste der *Remote Targets* hinzugefügt werden:

```
hosts -R
```

Mit dem Befehl `exit` kann *Metasploit* beendet werden.

Hinweis zur Verwendung von *Metasploit* – 2. Module

Metasploit besteht aus einer Sammlung von kategorisierten Modulen, in die mit dem Befehl `use`, gefolgt von einem Leerzeichen und dem Modulnamen, gewechselt werden kann. Mit dem Befehl `back` kann das Modul wieder verlassen werden.
Mit dem Befehl `search`, gefolgt von einem Leerzeichen und einem Suchbegriff, ist es möglich, Module zu suchen. Als Suchergebnis wird eine Liste von Modulen zurückgeliefert. Zum Beispiel liefert

```
search mozilla
```

alle Module, die irgendeinen Bezug zu *Mozilla*-Browsern haben. Da die Suche nicht immer einwandfrei funktioniert, wird empfohlen, die Autovervollständigung mit der Tabulatortaste zu verwenden – gibt man beispielsweise

```
use exploit/windows/browser/mozilla_
```

(ohne Bestätigung mit der Enter-Taste) ein und drückt die Tabulatortaste, erscheint eine Liste von Vorschlägen, die durch weitere Eingabe verkürzt werden kann. Verbleibt nur noch eine Möglichkeit, wird diese automatisch komplett vervollständigt.
Ist in ein Modul gewechselt worden, z.B. mit

```
use exploit/windows/browser/mozilla_firefox_onreadystatechange
```

können mit den Befehlen `info` und `options` Modulinformationen bzw. eine Liste der unterstützten Modulparameter angezeigt werden. Alle Parameter, die als notwendig (*yes* in der Spalte *Required*) markiert sind, müssen gesetzt werden. Das Setzen erfolgt nach dem Muster `set <Parametername> <Wert>`, z.B.

```
set SRVHOST 192.168.168.100
```

Nach dem Setzen aller Parameter kann das aktuelle Modul mit dem Befehl `run` gestartet werden. Einige Module unterstützen zusätzlich den Befehl `check`, mit dem **vor** dem Start überprüft werden kann, ob passende Ziele und/oder Anwendungsversion vorhanden sind.

Hinweis zur Verwendung von *Metasploit* – 3. Sitzungen

Mit *Metasploit* können mehrere aktive Sitzungen verwaltet werden. Der Befehl

```
sessions -l
```

listet alle aktiven Sitzungen nebst Typ auf. Analog beendet der Parameter `K` alle aktiven Sitzungen:

```
sessions -K
```

Mit dem Parameter `c`, gefolgt von einem Leerzeichen und einem Befehl, kann ebendieser Befehl an alle aktiven Sitzungen übertragen und ausgeführt werden, z.B.

```
sessions -c id
```

Außerdem ist es möglich, Befehle direkt in eine Sitzung einzugeben. Dazu muss mit

```
sessions -i <Sitzungsnummer>
```

(wobei Sitzungsnummer die Nummer aus der Auflistung mit `sessions -l` ist) in die Sitzung gewechselt werden (beachten Sie, dass manche Exploits automatisch Sitzungen öffnen). Anschließend können Befehle eingegeben werden. Der Befehl `background` schickt die Sitzung wieder in den Hintergrund – das funktioniert allerdings nur bei so genannten *meterpreter*-Sitzungen.
Der Typ einer Sitzung entscheidet darüber, welche Befehle unterstützt werden. Ein SSH-Login öffnet beispielweise eine Linux-Shell mit Benutzerrechten und Shellbefehlen (`id`, `ls` etc.). Eine *meterpreter*-Sitzung erlaubt hingegen weitgehenden Systemzugriff. „Reguläre“ Sitzungen können über Exploits in *meterpreter*-Sitzungen erweitert werden oder letztere zusätzlich öffnen. Der Exploit übernimmt dabei die „reguläre“ Sitzung als Parameter (meist `SESSION` mit der Nummer aus der Sitzungsauflistung, vgl. oben).

Hinweis zur Verwendung von *John the Ripper*

John the Ripper wird wie folgt aufgerufen:

```
john --format=<Passwortformat> <Dateipfad>
```

`<Dateipfad>` ist dabei der Pfad zu einer Datei, die Benutzernamen und (gehashte oder verschlüsselte) Passwörter enthält. `<Passwortformat>` spezifiziert das Format der Passwörter, z.B. `sha512crypt` für Passwörter, die mit SHA-512 gehasht worden sind und im Format von so genannten „unshadowed“ Linux-Passwortdateien vorliegen.
Im Standardmodus probiert *John the Ripper* zuerst eine Menge vordefinierter Standardpasswörter aus. Soll stattdessen ausschließlich Brute Force verwendet werden, kann *John the Ripper* mit dem zusätzlichen Parameter `--incremental: lanman` angewiesen werden, alle Passwörter mit Buchstaben, Ziffern und einigen (häufig verwendeten) Sonderzeichen auszuprobieren. Kurze und typische Passwörter werden dabei zuerst versucht. Mit `Q` kann der Prozess abgebrochen werden; beim Druck einer anderen Taste wird der aktuelle Fortschritt angezeigt. Liegt keine Passwortdatei vor, kann eine solche Datei wie folgt erzeugt werden:

```
(echo -n 'user:'; mkpasswd -m sha-512 "<Passwort>") > <Dateipfad>
```

Der geklammerte Befehl erzeugt dabei zuerst mit `echo` eine Zeichenkette mit einem leeren Benutzernamen, aber ohne Zeilenumbruch (Option `n`). Das Werkzeug `mkpasswd` hängt daran den SHA-512-Hash des angegebenen Passwortes sowie einige weitere Informationen, die die finale Zeichenkette zu einer gültigen Zeile einer Linux-Passwortdatei machen. Abschließend wird die Ausgabe des geklammerten Befehles mit dem `>`-Operator in die angegebene Datei umgeleitet, d.h. gespeichert. Diese Datei kann als Eingabedatei für *John the Ripper* verwendet werden, wie oben beschrieben.