Übungsblatt 05 – Netzwerksicherheit

Voraussetzungen: Wie in den vorigen Übungen sollte Sie in der Lage sein ein eigenes statisches Netzwerk aufzubauen. D.h. Ihr Netzwerk sollten den Anforderungen des Übungsblattes 3 genügen – es gibt einen Router der zwei Subnetze verbindet.

Nachdem Ihr Netzwerk aufgebaut ist:

- Überprüfen Sie ob sich Ihre VMs untereinander erreichen können.
- Überprüfen Sie ob alle VMs den Uplink erreichen können und ob externe Adresse (d.h. IP-Adressen außerhalb des Labors) erreichbar sind.
- Überprüfen Sie ob die Namensauflösung funktioniert.

Aufgabe A – SSH Basics

Die folgenden Aufgaben stellen *openSSH* als Werkzeug für eine sichere Kommunikation zwischen Prozessen (oder Rechnern) vor.
Zunächst sollen Sie sich mit dem Umgang mit *SSH* vertraut machen, anschließend werden theoretische Konzepte aus den Hausaufgaben auf die Umsetzung in der Praxis geprüft.

1. Starten Sie *Wireshark*, sodass Sie den anfallenden Netzwerkverkehr analysieren können.
   a) Loggen Sie sich via *SSH* auf dem Uranus-Server (`uranus-ai.f4.htw-berlin.de`) ein! Achten Sie darauf, dass zu viele Fehlversuche dazu führen, dass der Server Ihren Client blockiert. Das ist gewollt, um Bruteforce-Angriffe auf schlecht geschützte Account zu verhindern.
   b) Erläutern Sie die Bedeuten der Authentizitätsabfrage des *SSH*-Servers („authenticity“).
   c) Welche Bedeutung hat der Fingerprint?
   d) Wie können Sie den Fingerprint prüfen? Mit welchem Programm können Sie sich diesen anzeigen lassen?
   Bspw.: *SHA256:KsUg4lOc91/iJBYFkQhxeI/YGkcnKV2uKUXFNP1ymiw root@xen (ECDSA)*)
   e) Starten Sie in *Wireshark* einen neuen Traffic-Mitschnitt auf dem Ethernet-Netzwerkinterface. Anschließend soll eine neue *SSH*-Session von einem anderen Rechner gestartet werden. Analysieren Sie auszugsweise die entsprechenden Pakete! Welche Teile sind im Plaintext lesbar, ab wann greift die Verschlüsselung?
   f) Sie müssen sich bis jetzt immer via Passwort authentifizieren, d.h. Ihr Login erfolgt aufgrund eines Passworts. Ist Ihr Passwort in einem der ersten Pakete zu finden? Wenn es nicht zu finden ist, wie konnten Sie sich dennoch erfolgreich anmelden? (Welcher kryptografische Mechanismus greift hier ein...)

g) Wenn Sie die entsprechenden *Wireshark*-Mitschnitte ausgewertet haben, ist Ihnen aufgefallen, dass dort ein „Key Exchange“ stattfindet. Welches kryptografische Verfahren wird dort verwendet und ist dies eine symmetrisches oder asymmetrisches Chiffrierverfahren?

2. Ermöglichen Sie nun den Login mittels *SSH* zum Uranus-Server **ohne** das Nutzerpasswort angeben zu müssen. Dies sollten Sie auch für den Verbindungsaufbau zwischen Hosts in Ihrem Netz vornehmen (Debians ohne GUI).
Achtung: Wenn Sie sich auf dem Uranus ohne Passwort anmelden wollen, muss eine bereits existierende *SSH*-Verbindung auf dem Uranus-Server vorhanden sein, da ihr Home-Directory erst im Anschluss gemountet wird und ihr hinterlegter Public-Key ansprechbar ist.
Sie können dabei wie folgt vorgehen:

   a) Welches Public-Key-Verfahren wollen Sie für eine passwortlose Verschlüsselung nutzen? Welche Länge soll Ihr Key haben? Überlegen Sie sich vor dem eigentlichen generieren, was sinnvoll ist.

   b) Generieren Sie sich ein *SSH*-Schlüsselpaar! Nutzen Sie hierfür die recherchierten Parameter aus Ihren Notizen.

   c) Beim generieren des Schlüssels werden Sie aufgefordert eine Passphrase einzugeben. Der Private-Key ist durch eine Passphrase geschützt, sodass dieser geheime Schlüssel nur von Ihnen geöffnet werden kann.
   Wie ist die Passphrase zu wählen? Was gilt es zu beachten?

   d) Verbinden Sie sich von Rechner zu Rechner ohne ein Passwort zu nutzen. D.h. Sie sollten über das eigene LAN hinaus auf eine andere VM via *SSH* zugreifen können. Sie könne sich hierfür ein neuen Nutzer anlegen (*useradd*).

   e) Auf einer der VMs soll die Sicherheit etwas erhöht werden durch Anpassen der Settings.

      - Setzen Sie die Anzahl der maximalen Login-Fehlversuche auf drei!
      - Erlauben Sie dem Nutzer *student* nur noch das Auflisten des Home-Verzeichnis, wenn er sich via SSH verbunden hat.
      - Erlauben Sie zugriffe nur noch aus dem LAN 172.16.X.Y.
      - Setzen Sie als Anmeldeverfahren *SSH* auf reine Public-Key-Kryptografie. Hat dies eventuell auch Nachteile?

Aufgabe B – SSH-Forwarding

3. Mit *SSH* können Sie beliebige TCP-Verbindungen über die verschlüsselte *SSH*-Verbindung „tunneln“. Somit wird es Ihnen möglich, Server zu erreichen, zu denen Sie ansonsten keinen direkten Zugriff hätten (bspw. weil sie hinter einer Firewall

befinden oder der Datenverkehr anderweitig gefiltert wird – Packet-Filtering).
Konfigurieren Sie das Port-Forwarding unter SSH – ermöglichen Sie dazu folgende Zugriffe:

a) Nehmen Sie ein lokales Port-Forwarding auf die Website der HTW Berlin vor. Hierzu soll ein *SSH*-Tunnel (Source-Port 8080) aufgebaut werden, der den Verkehr auf den Standard HTTP-Port 80 führt (Destination Port 80).

b) Ihre VM logt sich per *SSH* auf einem anderen VM *SSH*-Server ein und leitet den lokalen Port 2200 auf den Port 22 des dortigen Systems weiter. Danach sollten Sie sich mit *SSH* über den lokalen Port mit dem *SSH*-Server des fremden *SSH*-Server verbinden können.

c) Konfigurieren Sie ein Remote-Port-Forwarding – stellen Sie dazu eine *SSH*-Verbindung vom einer anderen VM zu Ihrer VM als *SSH*-Server her. Leiten Sie den Port 8880 des *SSH*-Server nun über Ihren Client zum Webserver der URL `www.htw-berlin.de` weiter. Im folgenden kann sich nun jede VM mit Ihrer VM auf Port 8880 verbinden, um die Webseite der HTW-Berlin zu besuchen.

Aufgabe C – ARP-Cache-Poisoning

Wie in den Hausaufgaben bereits zu erahnen war, dürfen sie nun ein wenig Unruhe in Ihren Netzwerken stiften!
Sie sollen in diesem Teil der Laborübung ein ARP-Spoofing des Routers übernehmen. Um dies zu erreichen, sollen sie den ARP-Cache so manipulieren, das sämtlicher Verkehr nicht mehr über den eigentlichen Router geleitet wird, sondern über einen Angreifenden Host.

1. Zunächst müssen sie Angreifen und Opfer in ihren Netzwerk auswählen. Der Angreifer kann nicht der Router sein.

2. Analysieren sie den ARP-Cache des anzugreifenden Systems.

3. Klonen sie den *freeBSD* Router, da dieser schon die meisten Voreinstellungen hat und über Wireshark verfügt!

4. Da der angreifende Host (geklonter Router) als „MITM-Router“ fungiert, muss auch hier das Routing aktiviert sein und eine Default-Route existieren. D.h. im Idealfall bekommt die angreifende Maschine noch ein zusätzliches Interface (Bridge-Mode oder NAT). Daher muss auf diesem Rechner neben Forwarding auch NAT umgesetzt werden.

5. Im Moodle liegt ein Python-Skript (C, Perl ebenso), welche für den Angriff genutzt werden kann. Bevor sie dies einsetzen: Schauen sie sich das Skript an. Wie muss dieses Skript ausgeführt werden? D.h. mit welchen Rechten und welchen Argumenten muss das Skript aufgeführt werden?

6. Führen sie das ARP-Cache-Poisoning mithilfe des Skripts durch.
7. Lassen sie sowohl auf dem angreifenden als auch angegriffenen System Wireshark mitlaufen.
8. Betrachten sie den ARP-Cache während des Angriffs, sowie einige Zeit nachdem Angriff.
9. Ziehen sie ein Fazit aus dem eben durchgeführten Angriff!