بسم الله الرحمن الرحيم

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گزارش پایانی کارآموزی

نام و نام خانوادگی کارآموز:

مهروالسادات نوحی

شماره دانشجویی:

۹۹۳۶۱۳۰۶۱

استاد کارآموزی:

دکتر افسانه فاطمی

سرپرست محل کارآموز:

حسام الدین محمودی

محل کارآموزی:

شرکت مهندسی طراح مبتکر آریا تدبیر

آدرس:

تهران، خیابان دکتر شریعتی، بالاتراز چهارراه شهید مطهری، کوچه مینو، پلاک ۱۴، واحد۲

تلفن:

۰۲۱-۸۸۴۰۳۰۴۵

تاریخ شروع: ۱۴۰۲/۰۴/۱۷ **تاریخ پایان: ۱۴۰۲/۰۶/۱۷**

فهرست مطالب

عنوان **صفحه**

فهرست مطالب ۳
فهرست تصاویر ۴

چکیده ۵

فصل اول معرفی محل کارآموزی ۶

۱-۱ مقدمه ۶
۱-۲ معرفی پروژه ۶
۱-۳ اطلاعات محل کارآموزی ۷
۱-۴ آدرس و اطلاعات تماس ۷

فصل دوم پیش‌زمینه ۸

۲-۱ مقدمه ۸
۲-۲ سیستم عامل‌های شبکه ۸
۲-۳ سیستم عامل میکروتیک ۹
۲-۴ سیستم عامل اوبونتو ۱۱
۲-۵ ماشین مجازی ۵۱
۲-۶ نتیجه‌گیری ۱۳

فصل سوم کارهای مرتبط ۱۵

۳-۱ مقدمه ۱۵
۳-۲ پردازش بسته RADIUS ۱۵
۳-۳ بررسی ساختار RADIUS-HEADER ۱۶
۳-۴ بررسی ساختار RADIUS-ATTRIBUTE ۱۸
۳-۵ ایجاد بسته ACCEPT-REQUEST و ارسال به RADIUS-CLIENT ۲۴
۳-۶ ایجاد بسته REJECT-REQUEST و ارسال به RADIUS-CLIENT ۲۴

فصل چهارم ارزیابی کارآموزی ۲۵

۴-۱مقدمه ۲۵
۴-۲ تجربه‌ها ۲۵
۴-۳ ناکامی یا مشکلات ۲۶
۴-۴ ارزیابی کلی محل کارآموزی ۲۶
۴-۵ کتابخانه‌ها و پکیج‌ها پروژه ۲۶
فهرست مراجع ۲۸

فهرست تصاویر

شکل ۱- مدل Client-Server ۹
شکل ۲- نرم افزارwinbox صفحه ورود ۱۰
شکل ۳- نرم افزارwinbox صفحه اصلی ۱۰
شکل ۴- ایجاد ماشین مجازی‌های مجازی ۱۱
شکل ۵- ایجاد ماشین مجازی برای اوبونتو ۱۲
شکل ۶ - ایجاد ماشین مجازی برای میکروتیک ۱۲
شکل ۷- تنظیمات ماشین مجازی Ubuntu ۶۴-bit ۱۳
شکل ۸- تنظیمات ماشین مجازی میکروتیک ۱۳
شکل ۹- پرینت IP میکروتیک ۱۴
شکل ۱۰- پرینت IP اوبونتو ۱۴
شکل ۱۱- ساختار پردازش بسته در لینوکس ۱۳
شکل ۱۲ - پردازش بسته در لینوک ۱۶
شکل ۱۳- ساختار بسته دریافتی برای احراز هویت ۱۶
شکل ۱۴- بررسی ساختار RADIUS-HEADER ۱۷
شکل ۱۵- بررسی ساختار RADIUS-ATTRIBUTE ۱۵
شکل ۱۶- ساختار بسته RADIUS لینوکس ۱۸
شکل ۱۷- ساختار بسته RADIUS لینوکس برای استخراج نام کاربری ۱۹
شکل ۱۸- مراحل استخراج MSCHAP -RESPONSE, MSCHAP-CHALLENGE ۱۹
شکل ۱۹- احراز هویت با MSCHAP۲ ..... ۲۰
شکل ۲۰- احراز هویت با MSCHAP۲ ..... ۲۰
شکل ۲۱- ساختار MSCHAP۲ ..... ۲۱
شکل ۲۲- ساختار CHALLENGE-HASH ۲۱
شکل ۲۳- ساختار NT-PASSWORD-HASH ۲۲
شکل ۲۴- ساختار CHALLENGE-RESPONSE ۲۲
شکل ۲۵- ساختار CHALLENGE-RESPONSE ۲۳
شکل ۲۶- الگوریتم DES در این پروتکل ۲۳
شکل ۲۷- بسته ACCEPT-REQUEST و ارسال به RADIUS-CLIENT ۲۴
شکل ۲۸- بسته REJECT-REQUEST و ارسال به RADIUS-CLIENT ۲۴
شکل ۲۹ - کتابخانه‌ها و پکیج‌ها پروژ ۲۶
شکل ۳۰- کتابخانه‌ها و پکیج‌ها پروژه ۲۷

چکیده

احراز هویت[1] در شبکه‌های کامپیوتری نقش تأثیرگذاری در جامعه امروزی ایفا می‌کنند. احراز هویت یکی از جنبه‌های اساسی و حیاتی در زمینه امنیت اطلاعات و فناوری اطلاعات است. دراین دوره کارآموزی، به مفاهیم اصلی و اهمیت احراز هویت در محیط‌های مختلف پرداخته می‌شود. احراز هویت به معنای تشخیص و اثبات هویت افراد یا دستگاه‌ها در یک سیستم یا شبکه[2] است. این شامل روش‌های متعددی مانند نام کاربری و رمز عبور، اسناد شناسایی بیومتریک[3]، کلیدهای امنیتی، و تکنیک‌های دیگر می‌شود.در پایان دوره، شرکت قادر به ارتقاء امنیت اطلاعات کاربران و اطمینان از صحت هویت افراد و دستگاه‌ها در سیستم‌های مختلف خواهد بود.

در این دوره یاد گرفته می‌شود چگونه سیستم‌های احراز هویت پیاده‌سازی شود و به اعتماد و امنیت کاربران و داده‌ها اهمیت داده شود. این شامل ایجاد یک محیط تست جهت شناسایی و اصالت‌سنجی کاربران، استفاده از پروتکل‌ها و تکنولوژی‌های مختلف احراز هویت و اجرای راه‌حل‌های امنیتی مناسب برای مقابله با تهدیدات امنیتی می‌شود. این دانش و مهارت‌ها در جلب فرصت‌های شغلی در زمینه امنیت اطلاعات و فناوری اطلاعات تأثیرگذار خواهند بود. در این گزارش، ابتدا به معرفی شرکت مهندسی طراح مبتکر آریا تدبیر و پروژه‌های آن پرداخته شده، سپس یک پیش‌زمینه در مورد شروع پروژه در دوره کارآموزی و ایجاد محیط تست برای احراز هویت کاربران ارائه شده است. در ادامه، به کارهای مرتبط با تنظیمات اولیه و راه‌حل‌های پیشنهادی جهت انجام پروژه و ارزیابی کارآموزی پرداخته شده است. در نهایت، یک خلاصه‌بندی از تجربیات و دانش به دست آمده در دوره کارآموزی اختصاص یافته است.

---

[1] Authentication

[2] Network

[3] Biometric identification documents

فصل اول
معرفی محل کارآموزی

۱-۱ مقدمه

شرکت مهندسی طراح مبتکر آریا تدبیر در بهار سال ۱۳۹۰ با ورود به عرصه IT[۱] فعالیت خود را آغاز کرد.شرکت مهندسی طراح مبتکر آریا تدبیر، حاصل سال‌ها تجربه و تفکر خلاق از مجموعه مختصصانی جوان بوده است که در سال ۱۳۹۳ با هدف ارائه خدمات تخصصی در زمینه دیجیتال مارکتینگ نظیر برنامه نویسی، طراحی سایت، سئو و بهینه سازی مشغول به کار است. هدف اصلی این مجموعه رسیدن به استانداردهای جهانی، به روزرسانی اطلاعات و همراه شدن با آخرین تغییرات طراحی وب‌سایت و بهینه‌سازی آن است.

۱-۲ معرفی پروژه

پروژه Radius-server [۲] یک سرویس احراز هویت کاربران است . Radius-server یک پروتکل امنیتی و سرویس مورد استفاده در شبکه‌های کامپیوتری است. این پروتکل به منظور احراز هویت و مدیریت دسترسی کاربران به شبکه‌ها، به خصوص در شبکه‌های مبتنی بر ورود از راه دور [۳] به کار می‌رود. این پروتکل یک پروتکل شبکه است که در لایه Application اجرا می شود. پروتکل RADIUS از یک Radius-server و Radius-client استفاده می کند. Radius-client (یا سرور دسترسی به شبکه) یک دستگاه شبکه است (مانندروتر[۴]، سوئیچ[۵]) که برای تأیید اعتبار (احراز هویت) کاربران استفاده می شود. Radius-client با درخواست به سرور، کاربران را احراز هویت می کند.[۱]

---

[۱] Information Technology

[۲] Remote Authentication Dial-In User Service

[۳] Remote Access

[۴] Router

[۵] Switch

۱-۳ اطلاعات محل کارآموزی

این شرکت به مدیریت حسام الدین محمودی در زمینه برنامه نویسی حوزه وب وشبکه، طراحی وب سایت، طراحی اپلیکیشن موبایل مشغول به کار است. از محصولات و خدمات این شرکت صنعتی می‌توان به ارائه نرم افزارهای یکپارچه، دیجیتال مارکتینگ [۱]و.. اشاره کرد. نوع این شرکت به صورت سهامی خاص می‌باشد و تعداد نیروی فنی مشغول به کار به صورت تقریبی ۲۵ نفر می‌باشد. از نمونه کارهای این شرکت میتوان به موارد زیر اشاره کرد:[۲]

- فروشگاه ورزشی جیم لاین (سایت فروشگاهی)
- فروشگاه آنلاین مزون ایکس (سایت فروشگاهی)
- توسعه پژوهان (سایت آموزشی)
- داروخانه مرکزی اهواز (سایت فروشگاهی)

از خدمات مهم این شرکت میتوان به موارد زیر اشاره کرد:

- طراحی سایت
- خدمات سئو و بهینه سازی سایت
- واحد طراحی
- تبلیغات اینترنتی

۱-۴ آدرس و اطلاعات تماس

شرکت مهندسی طراح مبتکر آریا تدبیر،به آدرس تهران، خیابان دکتر شریعتی، بالاتراز چهارراه شهیدمطهری، کوچه مینو، پلاک ۱۴، واحد۲ واقع شده است. این شرکت با کد پستی ۱۶۳۹۶۴۸۵۵۳ و به آدرس

---

[۱] Digital Marketing

وبسایت و دارای پست الکترونیکی internship@ariatech.online و تلفن تماس ۰۲۱۸۸٤۰۳۰٤٥می‌باشد.

فصل دوم

پیش‌زمینه

۲-۱ مقدمه

شبکه هنگامی معنا پیدا می‌کند که سیستم‌ها بتوانند از طریق ارسال و دریافت داده با یکدیگر در ارتباط باشند. ارسال و دریافت داده باید از دو سیستم مجزا امکان پذیر باشد. سیستم‌عامل‌ها [۱]نقش بسیار مهمی در مفهوم و عملکرد شبکه‌ها دارند. در برخی موارد، سیستم‌عامل‌ها می‌توانند ویژگی‌های مدیریت شبکه را ارائه دهند. این ویژگی‌ها شامل مانیتورینگ شبکه، تنظیمات شبکه، و ابزارهای مدیریتی مرتبط با شبکه می‌شوند. به طور کلی، سیستم‌عامل‌ها به عنوان یک لایه میانی بین سخت‌افزار و نرم‌افزار در شبکه‌ها عمل می‌کنند و مدیریت، امنیت، و اجرای عملیات مختلف را در سیستم‌های کامپیوتری و شبکه‌ها فراهم می‌کنند. از این رو با بررسی سیستم‌های موردنظر در شبکه و نحوه ساختار آن دراین فصل اطلاعات بیشتری در ارتباط با تبادل اطلاعات سیستم‌ها و نحوه احراز هویت یک سیستم به سیستم دیگر حاصل خواهد شد.

۲-۲ سیستم عامل‌های شبکه

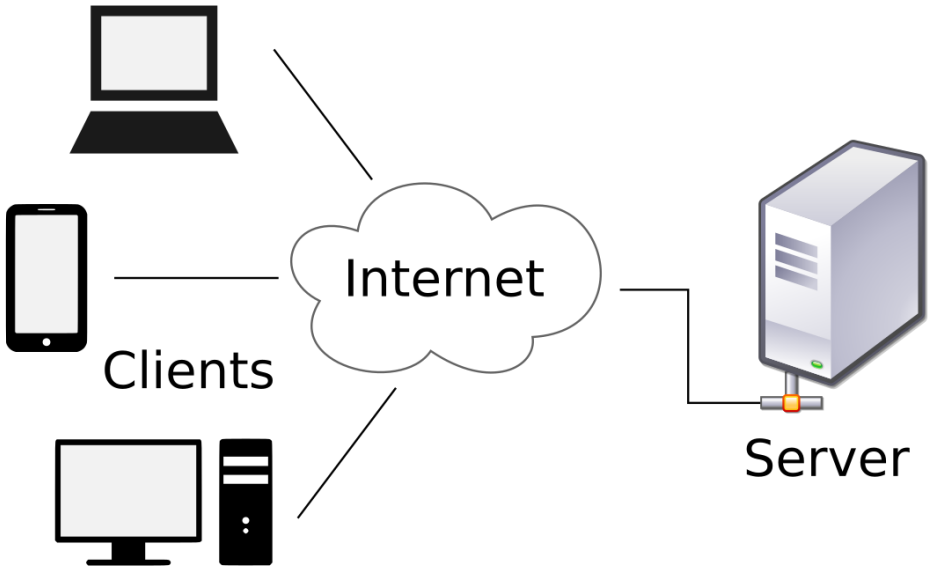
در این پروژه از دو سیستم عامل جهت سیستم‌های مجزا و تست احراز هویت کاربران استفاده شده است. از این رو اطلاعاتی در مورد سیستم عامل‌ها و نقش آن‌ها در حوزه شبکه خالی از لطف نیست. سیستم عامل در حوزه شبکه یک نقش بسیار مهم و حیاتی دارد. این سیستم‌ها به عنوان واسطه‌ای بین سخت‌افزار (مانند کامپیوترها، سرورها، روترها و دیگر تجهیزات شبکه) و نرم‌افزارها (برنامه‌ها و سرویس‌های شبکه) عمل می‌کنند. سیستم عامل‌های شبکه در واقع سیستم عامل‌هایی هستند که به جهت استفاده بر روی شبکه و سرویس دهنده‌ها و برای بهبود عملکرد شبکه تعبیه شده‌اند. در واقع سیستم عامل‌های شبکه تنها شامل ابزارهای شبکه‌ای موجود در

---

[۱] Operating Systems

سیستم عامل‌های کامپیوتری مانند لینوکس نبوده و نرم افزاری است که دسترسی کاربران، صف پیغام ها و ترافیک را کنترل و مدیریت میکند.[۳]

در این پروژه از دو سیستم عامل اوبونتو و میکروتیک استفاده شده است. همانطور که در بالا اشاره شد ما نیازمند ساختارClient-Server Model هستیم که در اینجا میکروتیک نقش سرور و اوبونتو نقش درخواست کننده را داراست.در ابتدا یک توضیحی در مورد هر دو سیستم عامل داده و چگونگی نقش آن‌ها را توصیف خواهد شد.



شکل ۱- مدل Client-Server

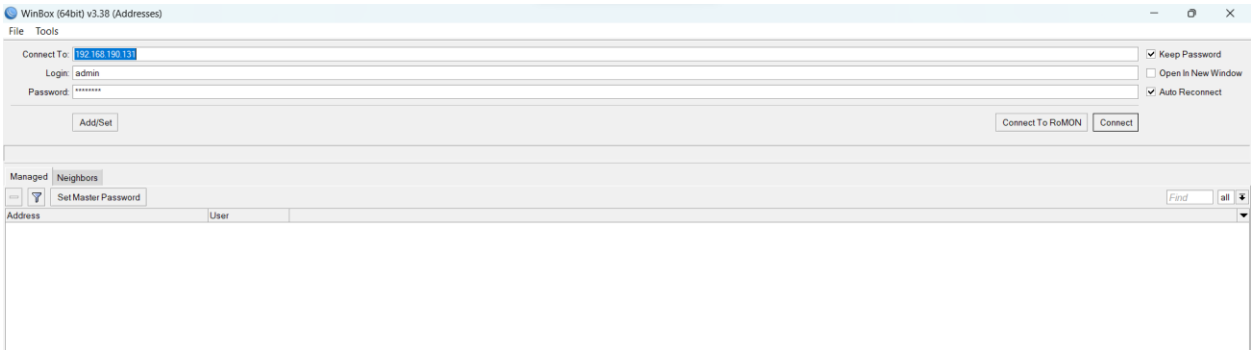
۲-۳ سیستم عامل میکروتیک

میکروتیک [۱]یک سیستم عامل مبتنی بر لینوکس است که با نام MikroTik RouterOS شناخته می‌شود و با نصب روی سخت افزار اختصاصی یا رایانه های مبتنی بر x۸۶ استاندارد، سخت افزار را به یک روتر شبکه تبدیل می‌کند و قابلیت‌های زیادی را در اختیار کاربر قرار می‌دهد. یکی از ویژگی‌های میکروتیک پایداری آن است؛

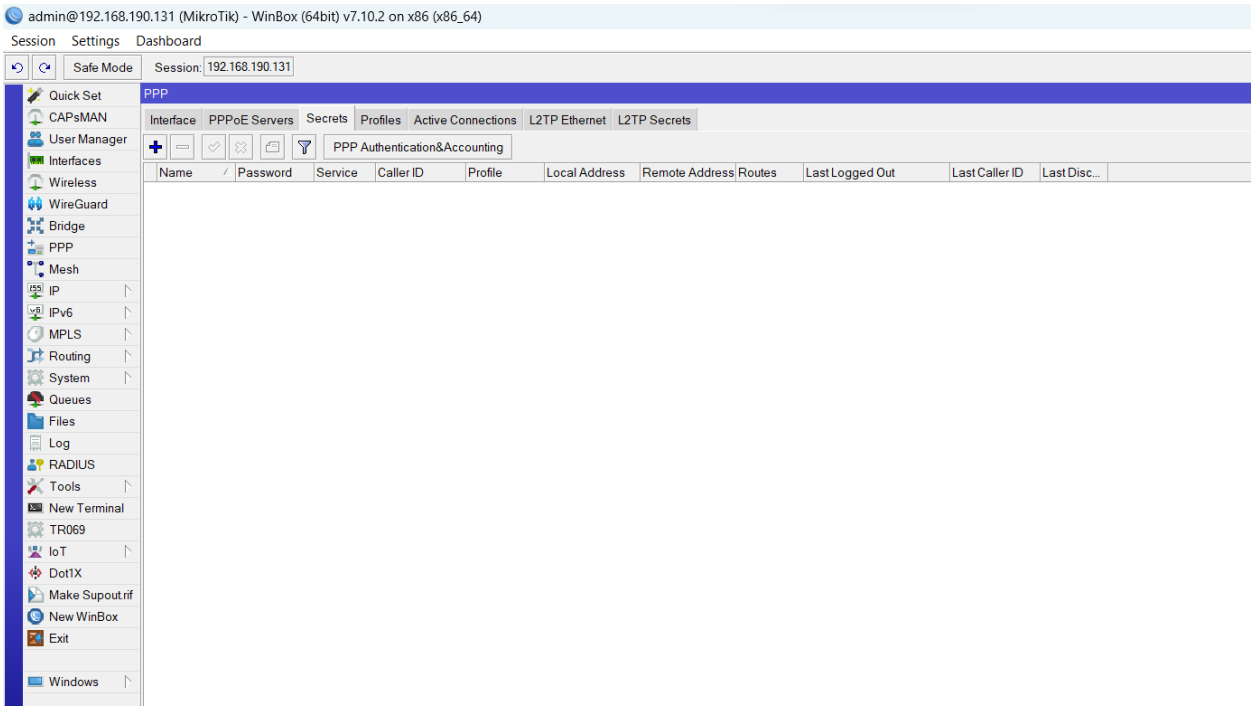
[۱] MikroTik

میکروتیک یک روتر قدرتمند از نظر سرعت و کارایی بالا می‌باشد که به راحتی در عرض چند دقیقه نصب و پیکربندی می‌شود و مزیت اصلی این روتر مقرون به صرفه بودن آن نسبت به مدل‌های سخت افزاری مشابه است.[٤]

winebox برنامه‌ای است که به کاربران این امکان را میدهد که به کمک آن به روتر میکروتیک متصل شوند. علاوه بر این به منظور مدیریت سرور مجازی میکروتیک نیز مورد استفاده قرار میگیرد .در حقیقت این نرم افزار یک رابط کاربری گرافیکی می‌باشد که به منظور کانفیگ و یا تنظیم روترها و نظارت بر نوع عملکرد آن‌ها مورد استفاده قرار می‌گیرد.



شکل ۲- نرم افزار winbox صفحه ورود



شکل ۳- نرم افزار winbox صفحه اصلی

۲-۴ سیستم عامل اوبونتو

اوبونتو [۱] یک توزیع لینوکس محبوب و مبتنی بر دبیان [۲] است. اوبونتو به عنوان یک سیستم عامل مبتنی بر لینوکس و با مجوز منبع باز توسعه یافته و توزیع می‌شود. این سیستم عامل به وسیله شرکت Canonical Ltd حمایت می‌شود و به عنوان یک سیستم عامل کاربری و سرور محبوب شناخته می‌شود. همه نسخه‌های اوبونتو می‌تواند به تنهایی بر روی یک کامپیوتر یا در یک ماشین مجازی اجرا شوند.

۲-۵ ماشین مجازی

ماشین مجازی [۳] یک سیستم کامپیوتری است که به وسیله نرم‌افزار شبیه‌سازی به نظر می‌آید و از دستگاه سخت‌افزاری مجازی بهره می‌برد. این دستگاه سخت‌افزاری مجازی به عنوان یک محیط مستقل عمل می‌کند و اجازه اجرای یک سیستم عامل مستقل بر روی یک دستگاه سخت‌افزاری واقعی را می‌دهد. یکی از مهم‌ترین

---

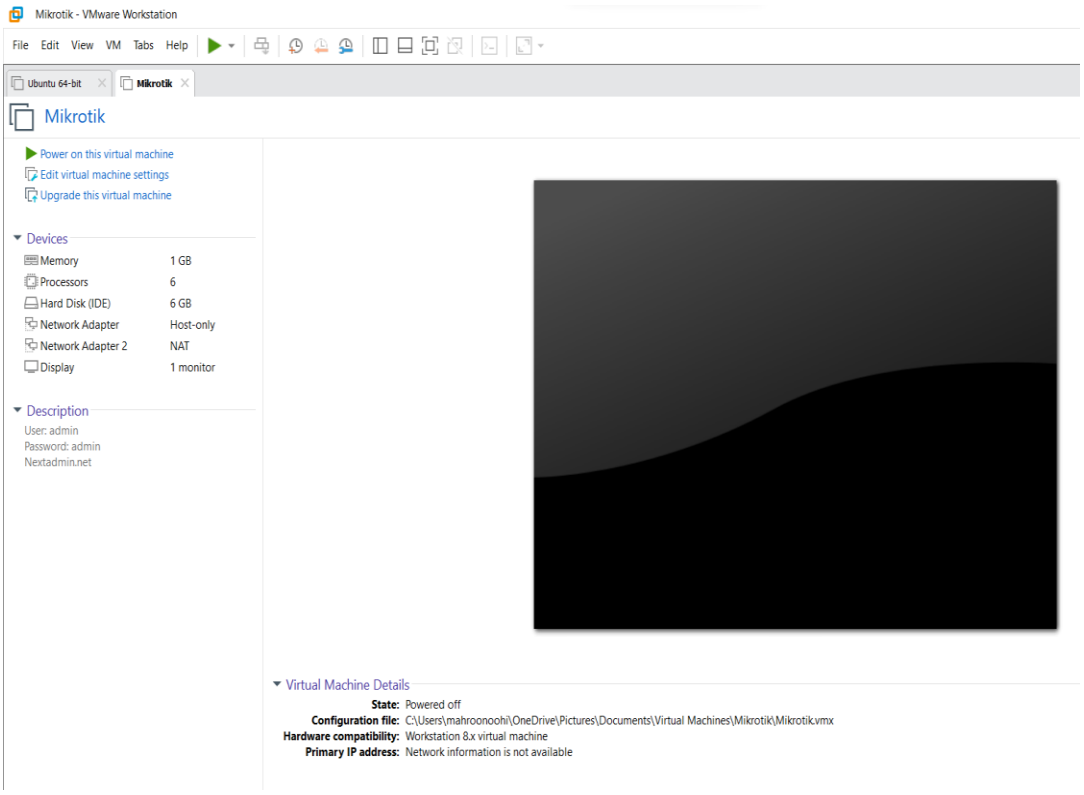
[۱] Ubuntu

[۲] Debian

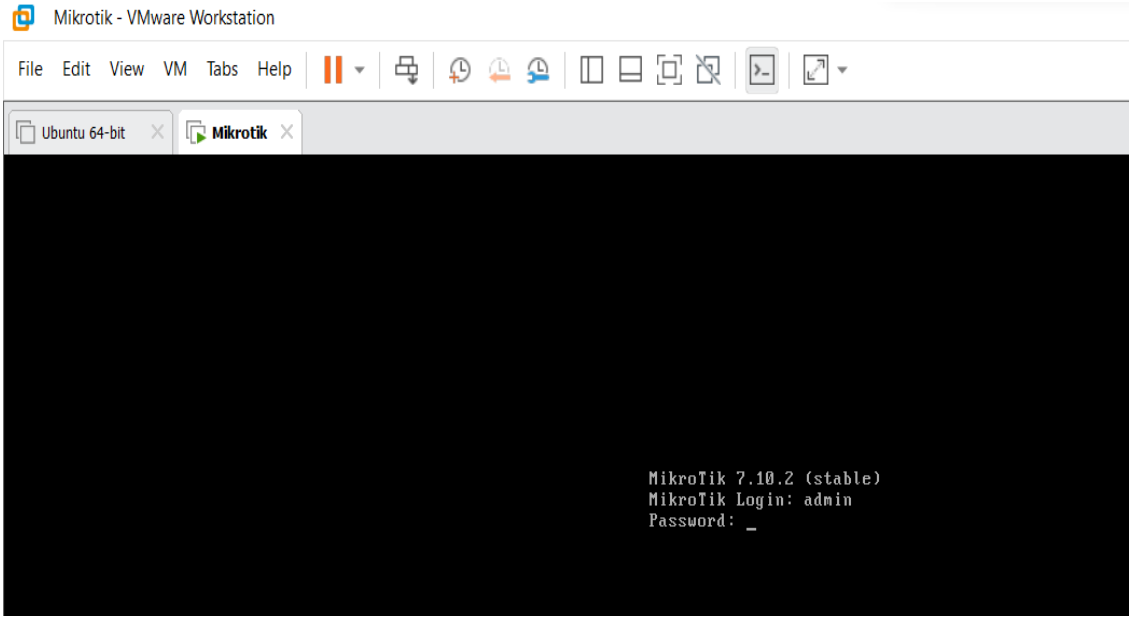
[۳] Virtual Machine

ویژگی‌های ماشین‌های مجازی ایزوله بودن آنهاست. هر VM مستقل از دیگر VMها عمل می‌کند و تداخل‌ها و تاثیرات جانبی بین VMها به حداقل می‌رسد. برای ایجاد ماشین مجازی، معمولاً از نرم‌افزارهای مدیریت ماشین‌های مجازی مانند VMware، VirtualBox، Hyper-V استفاده می‌شود. این نرم‌افزارها به ما امکان می‌دهند VMها را ایجاد، پیکربندی و مدیریت کنید. در این پروژه از ماشین مجازی VMware برای بالا آوردن سیستم عامل میکروتیک و اوبنتو استفاده شده است. برای هر سیستم عامل ماشین مجازی VM جدا در نظر گرفته شده است . در تصاویر تمامی تنظیمات برای بالا آوردن سیستم عامل‌ها بر روی هر ماشین مجازی قرار داده شده است.[۵]

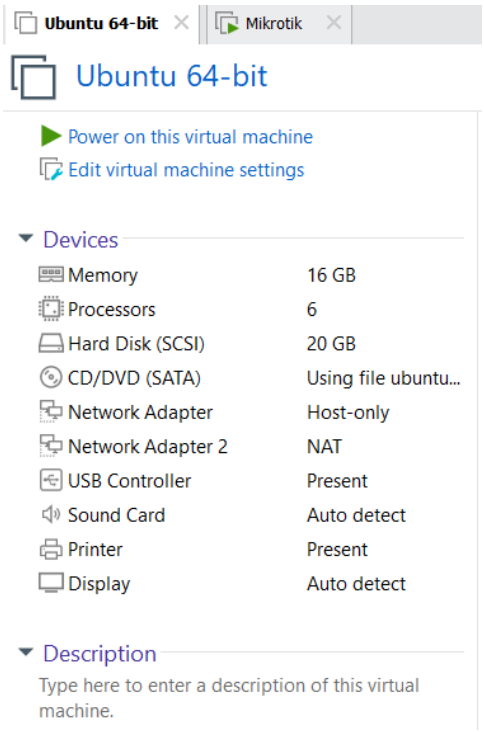
شکل ۴- ایجاد ماشین مجازی‌های مجازی



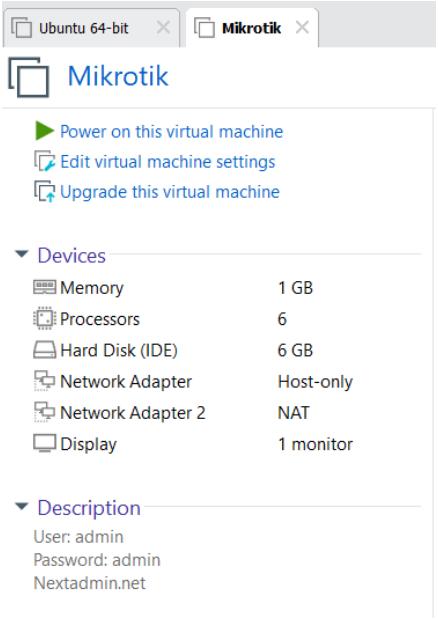
شکل ۵- ایجاد ماشین مجازی برای اوبونتو



شکل ۶ - ایجاد ماشین مجازی برای میکروتیک



شكل ٧– تنظيمات ماشين مجازى Ubuntu ٦٤-bit



شكل ٨- تنظيمات متشين مجازى ميكروتيک

۲ نتیجه‌گیری

بعد از ایجاد ماشین مجازی‌ها باید یک شبکه خصوصی بین آنها ایجاد تا بتوانند با یکدیگر ارتباط برقرار کنند از این رو در بخش تنظیمات Network-Adaptor با قرار دادن روی Host-only این کار میسر شدو اگر این دوسیستم در یک شبکه خصوصی بودند وبا داشتن IP باید می‌توانستند یکدیگر را پینگ بکنند زیرا دریک شبکه بودند و در مرحله پکیج‌های مورد نیاز جهت برنامه نویسی مانند PyCharm و پایتون اضافه گردید.

```
(31 messages not shown)
2023-09-09 00:21:42 system,error,critical router was rebooted without proper shu
tdown
2023-09-09 03:26:54 system,error,critical router was rebooted without proper shu
tdown
2023-09-09 03:30:32 system,error,critical router was rebooted without proper shu
tdown
2023-09-09 03:30:55 system,error,critical login failure for user admin via local
2023-09-12 02:32:03 system,error,critical router was rebooted without proper shu
tdown
2023-09-12 03:00:53 system,error,critical router was rebooted without proper shu
tdown
2023-09-12 03:11:44 system,error,critical router was rebooted without proper shu
tdown
2023-09-12 03:12:03 system,error,critical login failure for user admin  via loca
l

[admin@MikroTik] > /ip address print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
#   ADDRESS            NETWORK        INTERFACE
0 D 192.168.190.131/24 192.168.190.0  ether1
1 D 192.168.16.132/24  192.168.16.0   ether2
[admin@MikroTik] >
```

شکل ۹- پرینت IP میکروتیک

```
mahroo@mahroo-virtual-machine: ~
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:7c:aa:46 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.190.128/24 brd 192.168.190.255 scope global dynamic noprefixrou
te ens33
       valid_lft 1766sec preferred_lft 1766sec
    inet6 fe80::2c3f:4690:18d9:67a6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:7c:aa:50 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.16.130/24 brd 192.168.16.255 scope global dynamic noprefixroute
 ens37
       valid_lft 1766sec preferred_lft 1766sec
    inet6 fe80::4c80:78bb:5cf4:fca8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

شکل ۱۰- پرینت IP اوبونتو

فصل سوم
کارهای مرتبط

۳-۱ مقدمه

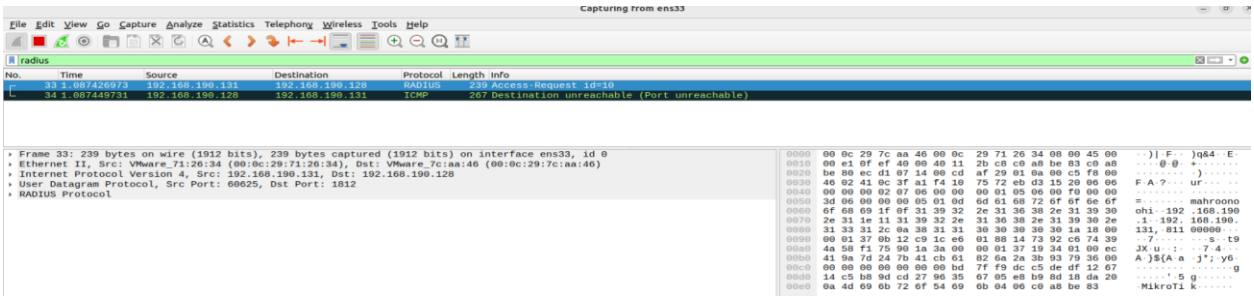
در این بخش یک سری تنظیمات اولیه برای مشاهده بسته‌های دریافتی از میکروتیک به اوبونتو در شبکه خصوصی انجام گرفت. در ابتدا با ساخت یک Radius-client که همان میکروتیک ما بود با درخواست به اوبونتو که Radius-server بود بسته درخواستی را ارسال کرده و منتظر پاسخ احراز هویت کاربر از سمت اوبونتو بود. در واقع کاربر با درخواست به میکروتیک (درخواست کننده) از طریق ویندوزconnection، و میکروتیک با درخواست به اوبونتو(سرور) منتظر پاسخ درخواست خود بود. در این حالت کاربر با ارسال نام کاربری و رمز عبور به میکروتیک منتظر پاسخ وی و میکروتیک با ارسال این بسته به اوبونتو احراز هویت صورت می‌گرفت. با نصب برنامه Wireshark روی اوبونتو وکار با آن برای مشاهده بسته‌های دریافتی واعمال فیلتر بر روی بسته‌های Radius بسته های درخواست ازسمت میکروتیک مشاهده شد .[٦]

۳-۲ پردازش بسته Radius

شبکه هنگامی معنا پیدا می‌کند که سیستم‌ها بتوانند از طریق ارسال و دریافت داده با یکدیگر در ارتباط باشند. این ارتباطات باید ساختارمند و از قوانین خاصی پیروی کنند. برای مثال اگر قرار باشد دو سیستم از طریق پروتکل UDP اقدام به رد و بدل کردن داده کنند، باید بسته‌های آن‌ها ساختار خاص UDP را داشته باشند، در غیر اینصورت بسته‌های آن‌ها امکان بازگشایی ندارند و بسته‌ی بدشکل تلقی می‌شوند. برای دریافت بسته و پردازش آن در اوبونتو از برنامه نویسی سوکت استفاده شده است.

```
87  def main():
88      """
89      Main function to start the RADIUS server and process incoming packets.
90      """
91
92      with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as udp_socket:
93          udp_socket.bind(('192.168.190.128', RADIUS_PORT))
94          print("Listening for packets on port 1812...")
```

شکل ۱۱- پردازش بسته در لینوکس



شکل ۱۲ - ساختار بسته دریافتی برای احراز هویت

```
▸ Frame 33: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits) on interface ens33, id 0
▸ Ethernet II, Src: VMware_71:26:34 (00:0c:29:71:26:34), Dst: VMware_7c:aa:46 (00:0c:29:7c:aa:46)
▸ Internet Protocol Version 4, Src: 192.168.190.131, Dst: 192.168.190.128
▸ User Datagram Protocol, Src Port: 60625, Dst Port: 1812
▾ RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0xa (10)
    Length: 197
    Authenticator: f8004602410c3fa1f4107572ebd31520
  ▾ Attribute Value Pairs
    ▸ AVP: t=Service-Type(6) l=6 val=Framed(2)
    ▸ AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
    ▸ AVP: t=NAS-Port(5) l=6 val=15728640
    ▸ AVP: t=NAS-Port-Type(61) l=6 val=Virtual(5)
    ▸ AVP: t=User-Name(1) l=13 val=mahroonoohi
    ▸ AVP: t=Calling-Station-Id(31) l=15 val=192.168.190.1
    ▸ AVP: t=Called-Station-Id(30) l=17 val=192.168.190.131
    ▸ AVP: t=Acct-Session-Id(44) l=10 val=81100000
    ▸ AVP: t=Vendor-Specific(26) l=24 vnd=Microsoft(311)
    ▸ AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
    ▸ AVP: t=NAS-Identifier(32) l=10 val=MikroTik
    ▸ AVP: t=NAS-IP-Address(4) l=6 val=192.168.190.131
```

شکل ۱۳- ساختار بسته دریافتی برای احراز هویت

همان طور که در شکل قابل مشاهده است بسته از پروتکل RADIUS از یک معماری سرویس گیرنده – سرویس دهنده برای تائید و accounting استفاده می‌نماید.درپروتکل RADIUS از پورت های ۱۸۱۲ برای احراز هویت و از ۱۸۱۳ برای Accountingاستفاده می‌شود. حال برای احراز هویت کاربر باید نام کاربری و رمز عبور وی را استخراج کرده و با اطلاعات داخل Radius-server چک شود و در صورت صحت یک پیغام مبنی بر صحت اطلاعات کاربر درست فرستاده شود و در صورت عدم صحت یک پیغام مبنی برای عدم درستی اطلاعات کاربر ارسال گردد. در بررسی ساختار بسته، بسته شامل Header و Attribute میباشد در ادامه توضیح داده خواهد شد.

۳-۳ بررسی ساختار Radius-header

یکی از اجزای پروتکل RADIUS است که برای ارسال اطلاعات مربوط به احراز هویت و مدیریت دسترسی به شبکه استفاده می‌شود. این هدر حاوی اطلاعات مهمی است که در ارتباط با فرآیند احراز هویت و اجازه دسترسی به منابع شبکه تعیین‌کننده است. بسته Radius دارای ۲۰ بایت هدر می‌باشد. ساختار اصلی هدر "Radius" به شکل زیر است:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Authenticator                         |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-
```

شکل ۱۴- بررسی ساختار Radius-header

- Code (کد): این فیلد یک بایتی است و نشان‌دهنده نوع پیام Radius است. مقادیر معمولی شامل مواردی مانند Access-Request (۱) برای درخواست احراز هویت، Access-Accept (۲) برای تایید دسترسی، Access-Reject (۳) برای رد کردن دسترسی و غیره هستند.

- Identifier (شناسه): این فیلد یک بایتی است و به عنوان شناسه منحصر به فرد درخواست و پاسخ‌های مرتبط با یکدیگر عمل می‌کند. این فیلد به ایجاد ارتباط بین درخواست و پاسخ‌های متعلق به یک تراکنش کمک می‌کند.

- Length (طول): این فیلد یک واحد دو بایتی است و نشان‌دهنده کل طول پیام Radius (هدر و محتویات) در بایت‌ها است.

- Authenticator (اعتبارسنج): این فیلد ۱۶ بایتی است و اطلاعات اعتبارسنجی برای پیام را در خود دارد. این اعتبارسنج برای ایجاد امنیت در پروتکل Radius استفاده می‌شود.

- Attributes (ویژگی‌ها): این بخش از پیام شامل ویژگی‌های مختلفی است که اطلاعات مربوط به احراز هویت و مدیریت دسترسی را حاوی می‌شوند. هر ویژگی شامل یک کد و طول مشخص و مقدار ویژگی می‌شود.

ساختار این هدر در تعیین نوع پیام و تبادل اطلاعات بین کلاینت و سرور RADIUS بسیار مهم است و امکان ارسال درخواست‌های احراز هویت و مدیریت دسترسی به منابع شبکه را فراهم می‌کند.

۴-۳ بررسی ساختار Radius-Attribute

پروتکل RADIUS برای انتقال اطلاعات مربوط به احراز هویت و مدیریت دسترسی به شبکه از ویژگی‌ها (Attributes)استفاده می‌کند. هر ویژگی دارای یک کدAttribute Type ، طول Length و مقدار Value است. ساختار یک ویژگی در پروتکل RADIUS به شکل زیر است:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |             Value ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

شکل ۱۵- بررسی ساختار Radius-Attribute

طبق RFC۲۸۵۶ برای استخراج نام کاربری کد ۱ تعبیه شده است برای استخراج کل اطلاعات بسته از تابع زیر استفاده شده است.

```
packet_type, packet_id, packet_length, authenticator = struct.unpack( __format: '!BBH16s', packet[0:20])

attributes = {
    "packet_type": packet_type,
    "packet_id": packet_id,
    "packet_length": packet_length,
    "authenticator": authenticator.hex(),
}
```

شکل ۱۶ - ساختار بسته Radius لینوکس

```python
# Parse AVPs
avp_start = 20
while avp_start < packet_length:
    avp_type, avp_length = struct.unpack( __format: '!BB', packet[avp_start:avp_start + 2])
    avp_value = packet[avp_start + 2:avp_start + avp_length]

    # Extract User-Name
    if avp_type == 1:
        attributes["User-Name"] = avp_value.decode('utf-8')
```

شکل ۱۷- ساختار بسته Radius لینوکس برای استخراج نام کاربری

اما برای امنیت ارسال رمز عبور، رمز عبور به صورت متن آشکار فرستاده نمی‌شود. بلکه از سیستم MSCHAP[۱] استفاده می‌شود. MSCHAP یک پروتکل احراز هویت است که به عنوان یکی از روش‌های احراز هویت در شبکه‌های کامپیوتری و ارتباطات استفاده می‌شود. این پروتکل اصولاً برای احراز هویت کاربران در شبکه‌های ویندوز مایکروسافت توسعه داده شده است. از این پروتکل به عنوان یک پروتکل احراز هویت نقطه به نقطه عمل می‌کند. این پروتکل از یک چالش (challenge) و یک پاسخ (response) برای احراز هویت استفاده می‌کند تا اطلاعات کاربری و رمزعبور را تأیید کند.

```python
        # Extract Vendor-Specific attributes
        elif avp_type == 26:
            vendor_id = struct.unpack( __format: '!I', avp_value[0:4])[0]
            if vendor_id == MICROSOFT_VENDOR_ID:
                vendor_type, vendor_length = struct.unpack( __format: '!BB', avp_value[4:6])
                vendor_value = avp_value[6:6 + vendor_length - 2]

                # Extract MS-CHAP-Challenge and MS-CHAP2-Response
                if vendor_type == 11:
                    attributes["MS-CHAP-Challenge"] = vendor_value
                elif vendor_type == 25:
                    attributes["MS-CHAP2-Response"] = vendor_value

        avp_start += avp_length
```

شکل-۱۸ مراحل استخراج MSCHAP -RESPONSE, MSCHAP-CHALLENGE

[۱] Microsoft Challenge Handshake Authentication Protocol

```python
def verify_mschapv2_response(parsed_packet, username, password):
    """
    Verify the MS-CHAPv2 response using the provided username and password.

    Args:
    - parsed_packet (dict): Parsed attributes from the RADIUS packet.
    - username (str): The username.
    - password (str): The password.

    Returns:
    - bool: True if the MS-CHAPv2 response is valid, False otherwise.
    """
```

شکل ۱۹- احراز هویت با MSCHAP۲

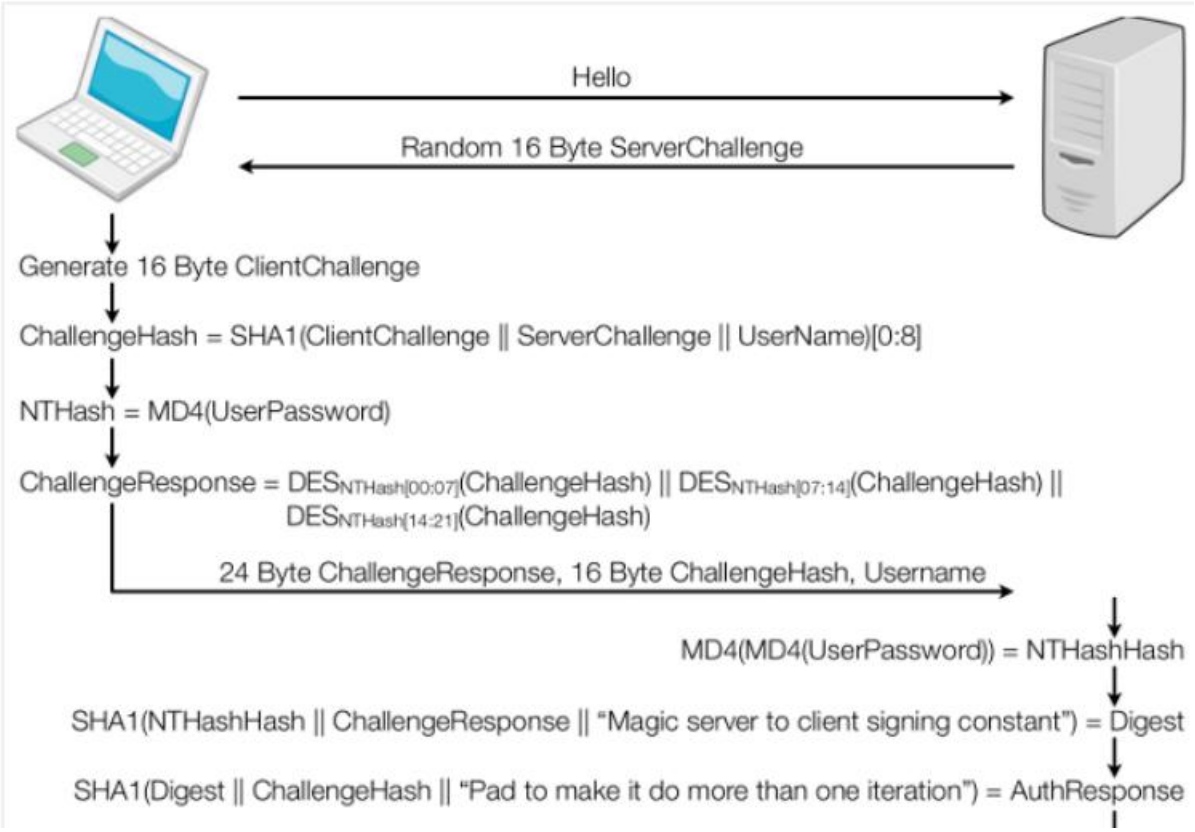
```python
    # Extract challenges
    peer_challenge = parsed_packet["MS-CHAP2-Response"][2:18]
    authenticator_challenge = parsed_packet["MS-CHAP-Challenge"]

    # Compute the expected MS-CHAPv2 response
    expected_response = mschap.generate_nt_response_mschap2(authenticator_challenge, peer_challenge, username,

    return expected_response.hex() == parsed_packet["MS-CHAP2-Response"][26:].hex()
```

شکل ۱۵- احراز هویت با MSCHAP۲

ساختار **MSCHAP-CHALLENGE**: [٦]



شکل ۱۶- ساختار MSCHAP۲

```
def challenge_hash(peer_challenge, authenticator_challenge, username: str):
    """ChallengeHash"""
    sha_hash = hashlib.sha1()
    sha_hash.update(peer_challenge)
    sha_hash.update(authenticator_challenge)
    sha_hash.update(bytes(username, encoding='latin1'))
    return sha_hash.digest()[:8]
```

شکل ۲۲ - ساختار Challenge-Hash

```python
def nt_password_hash(passwd):
    """NtPasswordHash"""
    pw = utils.str2unicode(passwd)
    md4_context = md4.new()
    md4_context.update(pw)
    return md4_context.digest()


def hash_nt_password_hash(password_hash):
    """HashNtPasswordHash"""
    md4_context = md4.new()
    md4_context.update(password_hash)
    return md4_context.digest()
```

شکل ۲۳ - ساختار NT-password-hash

```python
def generate_lm_response_mschap(challenge, password):
    password_hash = lm_password_hash(password)
    return challenge_response(challenge, password_hash)


def generate_nt_response_mschap(challenge, password):
    password_hash = nt_password_hash(password)
    return challenge_response(challenge, password_hash)


def generate_nt_response_mschap2(authenticator_challenge, peer_challenge, username, password):
    """GenerateNTResponse"""
    challenge = challenge_hash(peer_challenge, authenticator_challenge, username)
    password_hash = nt_password_hash(password)
    return challenge_response(challenge, password_hash)
```

شکل ۲۴- ساختار Challenge-Response

```python
def challenge_response(challenge: bytes, password_hash: bytes) -> bytes:
    """ChallengeResponse"""
    zpassword_hash = password_hash.ljust(__width: 21, __fillchar: b'\0')

    response = b""
    des_obj = des.DES(zpassword_hash[0:7])
    response += des_obj.encrypt(challenge)

    des_obj = des.DES(zpassword_hash[7:14])
    response += des_obj.encrypt(challenge)

    des_obj = des.DES(zpassword_hash[14:21])
    response += des_obj.encrypt(challenge)
    return response
```

شکل ۲۵- ساختار Challenge-Response

```python
def des_hash(clear: bytes):
    """DesEncrypt"""
    des_obj = des.DES(clear)
    return des_obj.encrypt(b"KGS!@#$%")
```

شکل ۲۶- الگوریتم DES در این پروتکل

۳-۵ ایجاد بسته Accept-request و ارسال به Radius-client

بعد از احراز هویت باید نتیجه به میکروتیک ارسال شود که آیا کاربر مجاز به لاگین هست یا خیر از این رو این بسته ساخته می‌شود تا صحت اطلاعات ارسال شده از سمت کاربر را به میکروتیک اعلام کند. ساختار بسته به صورت زیر می‌باشد.[۷]

Code

2 for Access-Accept.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Response Authenticator                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

شکل ۲۷- بسته Accept-request و ارسال به Radius-client

۳-۶ ایجاد بسته Reject-request و ارسال به Radius-client

اگر مقداری از ویژگی های دریافتی قابل قبول نباشد، پس سرور RADIUS باید بسته ای را با مجموعه فیلد کد ۳ (دسترسی-رد کردن)ارسال کند.

Code

3 for Access-Reject.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

شکل ۲۸- بسته Reject-request و ارسال به Radius-client

فصل چهارم
ارزیابی کارآموزی

۴-۱مقدمه

ارزیابی کارآموزی یک فرآیند مهم در توسعه مهارت‌ها و دانش افراد در محیط‌های کاری است. این ارزیابی به شرکت‌ها و سازمان‌ها کمک می‌کند تا عملکرد کارآموزان را ارزیابی کرده و از آنها برای بهبود عملکردشان استفاده کنند. ازاین رو دوره کارآموزی میتواند فرصتی طلایی را جهت حضور در محیط‌های صنعتی، تجاری و خدماتی در اختیار دانشجویان قرار دهد؛ لذا از این طریق دانشجو میتواند به کسب تجارب و مهارت‌های ارزشمند دست یابد. باتوجه به اهمیت گذراندن دوران کارآموزی و تأثیرات آن، در این فصل به طورکلی به بررسی و شرح تجربیات و دستاوردهای کسب شده در طول دوره کارآموزی در شرکت مهندسی طراح مبتکر آریا تدبیر و همچنین ارزیابی کلی این محل، پرداخته شده است.

۴-۲ تجربه‌ها

تجربه کارآموزی یکی از مراحل مهم در مسیر تحصیلی و حرفه‌ای یک فرد است. این تجربه به فرد فرصت می‌دهد تا مهارت‌ها و دانش خود را در محیط کاری واقعی تست کند و با مفاهیم و تجارب عملی ارتباط برقرار کند همچنین با شاخه‌ها و موضوعات مختلف و با تجربیات دیگران آشنا شود. در این دوره با مفاهیم مهم شبکه از قبیل احراز هویت و امنیت شبکه اطلاعات جدیدی حاصل گردید و از این احراز هویت با موفقیت انجام گردید. آشنایی با انواع نحوه رمزگذاری[۱]، انواع چالش‌ها و از این دست اطلاعات زیادی در اختیار فرد قرار می‌دهد. از آنجایی که با بررسی مقالات و تحقیق‌های زیاد به دانش فرد اضافه می‌گردد باعث پیشرفت وی خواهد شد. احراز هویت و امنیت شبکه اطلاعات موضوعاتی بسیار حیاتی در دنیای مدرن است. آشنایی با انواع نحوه رمزگذاری و چالش‌های مرتبط با امنیت شبکه اطلاعات، او را در اجرای پروژه‌های مختلف و حل مشکلات امنیتی تقویت می‌کند.علاوه بر این، تحقیق و مطالعه مقالات و تحقیقات مرتبط با حوزه امنیت شبکه و احراز هویت می‌تواند دانش فرد را بهبود بخشد و او را در جلب تازه‌ترین مفاهیم و تکنولوژی‌ها کمک کند.

---

[۱] Encryption

۴-۳ ناکامی یا مشکلات

یکی از مشکلات اساسی و مهم رخداده در دوران کارآموزی، نداشتن آشنایی کافی با دنیای شبکه، و عدم اطلاعات کافی در مورد نحوه رمزگذاری و پیچیدگی پروژه مورد نظر برای نداشتن آشنایی کافی با MSCHAP۲ جهت احراز هویت می‌باشد. از موارد دیگر آشنا نبودن با ساختار بسته زمان زیادی از فرد خواهد گرفت و باعث شد روی اصل موضوع به درستی تمرکز نشود.

۴-۴ ارزیابی کلی محل کارآموزی

به صورت کلی شرکت مهندسی طراح مبتکر آریا تدبیر شرکتی متشکل از افرادی حرفه‌ای و باتجربه بوده و تمامی آن‌ها در حوزه کاری خود مسلط و مسئولیت‌پذیر هستند. همچنین در تمامی مدت کارآموزی برخورد و رفتار تمامی اشخاص کاملاً محترمانه بود. این محل کارآموزی حتماً به دانشجویان متقاضی که به دنبال شرکتی متشکل از تیم حرفه‌ای و باتجربه هستند، و هدف آن‌ها کسب تجربه‌ای مفید از دوران کارآموزی خود است و از آنجایی که در زمینه‌های مختلف این شرکت فعالیت می‌کند برای دانشجویانی که به دنبال کسب تجربه در زمینه مختلف هستند پیشنهاد می‌شود.[۸]

۴-۵ کتابخانه‌ها و پکیج‌ها پروژه

```
import hashlib

from . import des
from . import md4
from . import utils
```

شکل ۲۹- کتابخانه‌ها و پکیج‌ها پروژه

```python
import socket
import struct

from py3mschap import mschap
from pyrad.dictionary import Dictionary
from pyrad.packet import AccessAccept, AuthPacket

# Constants
RADIUS_PORT = 1812
SECRET = b'11041104'
MICROSOFT_VENDOR_ID = 311
radius_packet_format = "!BBH16s"
```

شکل ۳۰- کتابخانه‌ها و پکیج‌ها پروژه

- از pyrad برای ایجاد بسته Accept-request و ارسال به Radius-client و Reject-request و ارسال به Radius-client استفاده شده است.

- از کتابخانه hashlib برای رمزگذاری استفاده شده است

فهرست مراجع

[١] Y. Ma and H. Ning, "Improvement of EAP Authentication Method Based on Radius Server," ٢٠١٨ IEEE ١٨th International Conference on Communication Technology (ICCT), Chongqing, China, ٢٠١٨, pp. ١٣٢٤-١٣٢٨, doi: ١٠,١١٠٩/ICCT.٢٠١٨,٨٦٠٠٠٧٧.

[٢] https://ariatech.online/

[٣] J. K. Ousterhout, A. R. Cherenson, F. Douglis, M. N. Nelson and B. B. Welch, "The Sprite network operating system," in Computer, vol. ٢١, no. ٢, pp. ٢٣-٣٦, Feb. ١٩٨٨, doi: ١٠,١١٠٩/٢,١٦.

[٤] Baskoro, Alexander Pandu. ٢٠١١. Analisis dan Perancangan Jaringan Komputer Studi Inna Garuda Yogyakarta. http://e-journal.uajy.ac.id/. Universitas Atma Jaya Yogyakarta, Mikrotik. [online] https://mikrotik.com/

[٥] VMWare. "VMware vSphere Hypervisor ٧,٠".

[٦] J. Feng, "Analysis, Implementation and Extensions of RADIUS Protocol," ٢٠٠٩ International Conference on Networking and Digital Society, Guiyang, China, ٢٠٠٩, pp. ١٥٤-١٥٧, doi: ١٠,١١٠٩/ICNDS.٢٠٠٩,٤٤.

[٧] P. Urien and M. Dandjinou, "Introducing Smartcard Enabled RADIUS Server," International Symposium on Collaborative Technologies and Systems (CTS'٠٦), Las Vegas, NV, USA, ٢٠٠٦, pp. ٧٤-٨٠, doi: ١٠,١١٠٩/CTS.٢٠٠٦.

[٨] “Google Meet” [Online]. Available: https://apps.google.com/meet