مروری ابتدایی بر حملات رایج علیه شبکه های بیسیم

قسمت اول : حملات علیه پروتکل های امنیتی شبکه

سید حمید کشفی

Hamid@OISSG.org

مهر ماه 1388

نگارش 1.1

فهرست مطالب :

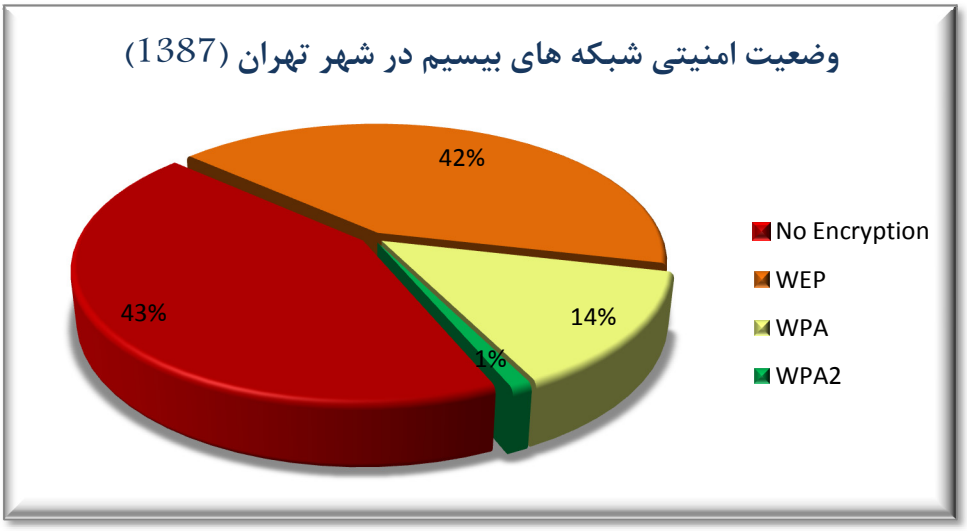
مقدمه:

سالها از زمان همه گیر شدن و عمومیت پیدا کردن استفاده از تکنولوژی شبکه های بیسیسم در سطح کاربران گذشته است و از آن زمان نیز ده ها تکنیک و روش حمله به سطوح و بخش های مختلف این تکنولوژی بصورت عمومی منتشر شده است. هدف اغلب این حملات و تکنیک ها در نهایت افشای اطلاعات و ترافیک شبکه و یا حصول دسترسی غیر مجاز به شبکه های حفاظت شده، به هر منظور می باشد. جالب اینجاست که با گذشت مدت زمان بسیار طولانی از اولین تحقیقات آکادمیک در این زمینه و تکمیل و پیاده سازی عملی حملات کشف شده، و افزایش چشمگیر سطح آگاهی کاربران در خصوص تهدیدات موجود علیه این بستر هنوز هم شبکه های راه اندازی شده بصورت نا امن بر بستر

بیسیم به وفور مشاهده شده و هنوز هم حلقه شکسته شده از زنجیره امنیت در بسیاری از سازمان ها و ارگان های خصوصی و دولتی دارای اطلاعات حساس که مورد نفوذ قرار میگیرند، بستر شبکه بیسیم می باشد. این مورد بخصوص در کشور ما نمود بسیار بیشتری داشته و قابل لمس تر است، بطوری که تجربه شخصی نویسنده در سال های اخیر نشان داده که بسیاری از دریافت کننده گان سرویس های امنیتی که اتفاقاً اهمیت زیادی نیز به موضوع کنترل دسترسی به شبکه داده و هزینه های گزافی برای سرویس ها، نرم افزارها و سخت افزارهای امنیتی پرداخت می کنند، علی رغم آگاهی از خطرات هنوز هم نسبت به تهدیدات موجود علیه شبکه های بیسیم بسیار آسیب پذیر هستند.

تحقیقات در خصوص شناسایی نقاط ضعف و روش های حمله به شبکه های بیسیم کماکان ادامه داشته و همواره شاهد این موضوع هستیم که با معرفی یک تکنولوژی و پروتکل امنیتی جدید برای این بستر که در پی تحقیقات آکادمیک مربوط به مشکلات امنیتی تکنولوژی های قدیمی تر بوده، روش های عملی (Practical) استفاده از این ضعف های امنیتی نیز با یک فاصله زمانی کشف و پیاده سازی شده و در قالب کد ها و ابزارهایی منتشر میگردند.

در یک بررسی فنی-آماری که بصورت تحقیق میدانی در سال 1387 در سطح شهر تهران بانجام رسیده مشخص شد که از مجموع حدود 5000 شبکه های بیسیم فعال شناسایی شده در دوره بررسی، 43٪ از شبکه ها بصورت کاملاً نا امن (عدم استفاده از هرگونه رمزنگاری در بستر شبکه) راه اندازی شده و از مجموع شبکه هایی که از روش های رمزنگاری استاندارد بستر استفاده کرده بودند نیز، 42٪ از شبکه ها از پروتکل هایی استفاده کرده اند که از نظر امنیتی منسوخ شده و براحتی مورد حمله واقع می شوند. پس در مجموع در حدود 85٪ از شبکه های شناسایی شده در آن دوره بررسی نسبت به تهدیدات امنیتی سطح پایین (از نظر دانش فنی مورد نیاز برای حمله) آسیب پذیر بودند و تنها 15٪ از کل شبکه های بیسیم شناسایی شده از پروتکل های امنیتی در سطح قابل قبول (در زمان انجام بررسی) استفاده کرده اند ! این آمار خود به تنهایی گویای وضعیت موجود و لزوم توجه بیشتر به این مقوله است. نتایج و جزئیات کامل بررسی مذکور بصورت عمومی منتشر نشده و در اختیار عموم قرار نخواهد گرفت.



در این مقاله سعی شده است تا با مروری بر روش های عمومی شناخته شده برای حمله به بسترهای شبکه بیسیم، ضمن آگاه کردن خواننده از تهدیدات موجود و برطرف کردن برخی از ذهنیت های اشتباه که در خصوص امنیت شبکه های بیسیم وجود دارد، وی را با به روش ها، تکنیک ها و ابزارهای اولیه ارزیابی امنیتی یک شبکه مبتنی بر تکنولوژی بیسیم آشنا سازد. این مقاله به هیچ عنوان یک منبع و مجموعه کامل

و یا پیشرفته از این اطلاعات نبوده و تشریح کامل این مبحث خارج از حوصله آن می باشد. سعی نویسنده بر این بوده است تا با گردآوری منابع و مطالب مختلف در این مقاله، نقطه شروعی را برای آغاز تحقیقات و فراگیری جزئیات بیشتر در اختیار خواننده قرار دهد. لازم به ذکر می باشد که فرض نویسنده بر این بوده که مخاطب مقاله آشنایی اولیه و کافی با تکنولوژی ها, پروتکل ها و اصطلاحات مورد استفاده در شبکه های بیسیم داشته و دید اولیه ایی در مورد مسائل پایه شبکه و سیستم عامل, امنیت آنها و رمزنگاری دارد.

مقدمات حمله

سخت افزارها و بستر مورد نیاز:

اولین قدم برای ارزیابی شبکه های بیسیم فراهم کردن بسترهای سخت افزاری مورد نیاز می باشد. انتخاب و استفاده از سخت افزارهای مناسب و صحیح از این جهت دارای اهمیت می باشد که در بسیاری از حملات که مبتنی بر نرم افزارهای خاص منتشر شده می باشند، پشتیبانی از سخت افزارهای مختلف محدود بوده و اگرچه پروژهای کد باز (Open Source) متعددی برای پشتیبانی از سخت افزارهای شبکه های بیسیم وجود دارند اما هنوز هم انتخاب پلتفورم مورد استفاده و سخت افزارهای پشتیبانی شده بدلیل محدودیت در استفاده از Device-Driver های خاص مشکل بسیاری از افرادی است که بدون تجربه قبلی اقدام به شروع یادگیری مطالب در این زمینه می کنند. در صورتی که پیش از این از کارت خود در محیط لینوکس استفاده کرده و با موفقیت کارت و ابزارهای رایج ارزیابی امنیت شبکه بیسیم را راه اندازی کرده و از آنها برای اتصال به شبکه بیسیم استفاده کرده اید می توانید بخش اول این مقاله را نادیده بگیرید.

(کارت شبکه بیسیم) : اولین آیتم از سخت افزارهای مورد نیاز، یک کارت مناسب برای دریافت سیگنال شبکه های بیسیم می باشد. فراگیر شدن استفاده از کامپیوترهای قابل حمل در بسیاری از موارد این نیاز اولیه را رفع کرده است بطوری که عموماً کارت های نصب شده برروی نوت بوک های موجود در بازار مبتنی بر چیپ ست هایی هستند که جوابگوی نیازهای اولیه ما می باشند. اگرچه کارت های تعبیه شده در نوت بوک ها (که بر روی اسلات های Mini-pci / Micro-pci) سیستم نصب شده در موارد و استفاده اولیه کارآمد بوده و به شما اجازه استفاده از بسیاری از ابزارها را می دهند، اما در شرایط واقعی حملات و استفاده حرفه ایی بسیاری از این کارت های داخلی محدودیت ها و مشکلات زیادی را سبب خواهند شد. اولین نکته منفی در استفاده از کارت های تعبیه شده در نوت بوک ها ضعف آنها در دریافت سیگنال می باشند، بطوری که در شرایط واقعی یک حمله در محیط خارج از آزمایشگاه میزان سیگنال دریافتی به اندازه کافی قوی نبوده و شما را ملزم به قرار گرفتن در یک محدوده فیزیکی بسیار نزدیک به شبکه هدف می نماید که در بسیاری از موارد امکانپذیر نیست. علت این ضعف در اغلب موارد ضعف خود کارت بیسیم و یا برند آن کارت نیست. تمامی کارت های بیسیم عرضه شده در بازار در هر شکلی، می بایست از یک استاندارد مشخص برای حداقل و حداکثر میزان توان تولیدی پیروی کنند. بطور مثال اکثر کارت های بیسیم نصب شده بر روی نوت بوک ها دارای توان خروجی بین 50 تا 200 میلی وات می باشند. البته در صورت تمایل شما به خرید کارت های Mini-PCI مدل هایی با توان خروجی بسیار بالاتر مانند 300, 400, 600 و حتی 1000 میلی وات نیز در دسترس هستند. در مورد کارت های PCMCIA با کیفیت نیز توان خروجی عموماً بین 300 تا 500 میلی وات است. حتی توان 100~200 میلی وات نیز در بسیاری از موارد و در صورت برقرار بودن سایر نیازمندی ها کافی است, اما اشکال کار در جای دیگریست. مشکل اصلی کارت های داخلی نوت بوک ها عدم وجود آنتن مناسب و قوی تعبیه شده در سیستم است. در بهترین حالت نوت بوک شما ممکن است مجهز به یک آنتن خارجی با گیرندگی 3 تا 4 دسی بل باشد که این میزان در مورد آنتن های داخلی (که عموماً در کنار و داخل قاب LCD تعبیه شده اند) با فرض با کیفیت بودن آنها حتی کمتر از این میزان است. مشکل دوم زمانی بروز خواهد کرد که شما قصد استفاده از یک آنتن خارجی را برای کارت های داخلی نوت بوک ها داشته باشید. برخلاف تصور عموم کارت های داخلی نیز امکان اتصال آنتن های خارجی را به شما می دهند. اشکال کار اینجاست که برای دستیابی به سوکت اتصال آنتن و

جایگزین کردن سیم آنتن داخلی می بایست اولاً نوت بوک باز شده (خدا حافظ گارانتی!) و ثانیاً محلی جدید برای عبور کابل آنتن خارجی و سوکت مربوط به آن بر روی بدنه نوت بوک فراهم شود و این کار هم در اکثر مواقع مساوی است با سوراخ کردن بدنه نوت بوک با مته، در صورتی که اصلاً جای خالی برای این سوراخ وجود داشته باشد. کمتر کسی حاضر به تقبل این دردسر و ریسک است. بنا بر این در صورت نا کافی بودن قدرت گیرندگی آنتن چاره ایی جز استفاده از کارت های جانبی PCMCIA/PCI Express ویا USB نیست.

در صورت تصمیم به استفاده از کارت های جانبی اولویت اول مجدداً امکان استفاده از آنتن جانبی بر روی کارت می باشد. در صورتی که قادر به تهیه کارت جانبی که سوکت اتصال آنتن بر روی آن بصورت استاندارد تعبیه شده است را ندارید بهتر است پول خود را هدر ندهید چون در صورت استفاده از آنتن خود کارت (اگر بتوان اصلاً آنرا آنتن نامید) تغییر چندانی را مشاهده نخواهید کرد. در صورتی که تصمیم به هک کردن کارت و اضافه کردن سوکت آنتن گرفته باشید نیز مجدداً سر و کار شما با ابزارآلات خواهد بود. جدای از این موضوع کارتهایی که بصورت پیشفرض سوکت اتصال آنتن خارجی ندارند، دارای توان خروجی کمتری نسبت به انواع دارای سوکت هستند و امکان استفاده از آنتن های قوی (High-Gain) برای آنها وجود ندارد. در نهایت انتخاب شما می بایست محدود به کارت های با چیپ ست مشخص باشد. کارت های مجهز به چیپ های Atheros و پس از آن برندهای Orinoco و Intel به ترتیب بهترین گزینه های عمومی و در دسترس موجود در بازار هستند. هر سه برند به خوبی توسط درایور هایی مانند MadWifi که در ادامه بیشتر در مورد آن توضیح داده خواهد شد پشتیبانی شده و اکثر کارت های مبتنی بر این سه چیپ و برند بدون مشکل خاصی توسط سیستم عامل شناسایی می گردند. بنا بر این در صورتی که نوت بوک شما مجهز به کارت داخلی از سری Intel می باشد (در صورت عدم لزوم استفاده از آنتن خارجی) شما با مشکل خاصی روبرو نخواهید شد. حتی در صورت متفاوت بودن چیپ کارت شما با مدل های یاد شده نیز شما با صرف چند دقیقه وقت برای جستجو براحتی قادر به یافتن دستورالعمل های لازم برای راه اندازی کارت خود در محیط لینوکس خواهید بود. مطمئن باشید در این زمینه خاص شما و مشکل شما اولین مورد اتفاق افتاده نیست و قبلاً توسط دیگران تجربه شده است. در خلال جستجو برای کارت و درایورهای سازگار (برای بررسی های امنیتی شبکه بیسیم، و نه کاربری معمولی) از دو مورد غافل نشوید؛ اولاً امکان فعال کردن حالت Promiscuous (که در مورد کارت های بیسیم به آن Monitor Mode گفته می شود) و ثانیاً امکان Packet Injection. در مورد هر دو امکان در ادامه مقاله به تفضیل توضیحاتی داده خواهد شد. کارت های بیسیم تحت نام های تجاری بسیار متنوع در دسترس هستند اما با مرور مشخصات فنی آنها می توانید نوع چیپ مورد استفاده را مشخص کنید. این چیپ استفاده شده در کارت است که ملاک پشتیبانی نرم افزاری از کارت است و نه لزوماً برند و مارک شرکت تولید کننده یک کارت. همچنین هر خانواده از چیپ ها نیز خود دارای کد های مشخص بوده که در انتخاب کارت و مقایسه آن با لیست سخت افزارهای پشتیبانی شده توسط نرم افزار یا درایور می بایست به آن دقت کرد. خود من همیشه از کارت های مبتنی بر Atheros استفاده کرده و نتایج قابل قبولی را نیز گرفته ام. Ubiquity یکی از برند های معتبر است که کلیه کارت های آن مبتنی بر چیپ های Atheros بوده و کارت های متنوع برای انواع درگاه ها را تولید می کند. Ubiquity تنها شرکت تولید کننده کارت های بیسیم مبتنی بر Atheros نیست، اما یکی از بهترین های این دسته از کارت ها می باشد. کارت هایی که من برای بررسی های مختلف بسته به نیاز از آنها استفاده می کنم کارت Mini-pci مدل Intel-PRO 2200، کارت PCMCIA مدل SRC Long-range و کارت PCIExpress مدل SR71-X می باشند. تمامی کارت های یاد شده Multi-band بوده و از پروتکل های A/B/G پشتیبانی می کنند. کارت SR71-X همچنین یکی از معدود کارت های قابل اطمینان Long-range است که از تکنولوژی 802.11n نیز پیشتیبانی می کند. برای دسترسی به لیستی از کارت ها و برند ها و آگاهی از جزئیات آنها می توانید به آدرس های زیر مراجعه کنید:

http://aircrack-ng.org/doku.php?id=compatible_cards

http://www.seattlewireless.net/index.cgi/HardwareComparison

http://madwifi-project.org/wiki/Compatibility

(آنتن مناسب) : دومین مورد در اختیار داشتن یک آنتن مناسب برای اتصال به کارت می باشد. در صورتی که شما از کارت های جانبی استفاده کنید برخی از این کارت ها در بسته بندی خود یک یا دو عدد آنتن را نیز عرضه می کنند. قدرت این آنتن ها بین 5 تا 7 دسی بل می باشد که برای مصارف داخلی و عمومی کافی است. این که چه نوع آنتن برای کار شما مناسب است ارتباط مستقیم با کاربرد و روش استفاده شما از آنتن دارد. آنتن های موجود در دو دسته کلی جای می گیرند. دسته اول آنتن های چند جهتی (Omni) و دسته دوم آنتن های جهت دار (Directional) می باشند. تفاوت اصلی این دو دسته در زاویه پوشش و دید آنتن می باشد. در صورتی که هدف شما بررسی عمومی و در فواصل نسبتاً نزدیک است بطوری که آنتن بدون نیاز به تنظیم و قرار گرفتن در زاویه و جهت خاص قادر به دریافت سیگنال باشد، استفاده از آنتن های Omni مناسب تر است. یک مثال این کاربرد استفاده از آنتن های Omni برای انجام Site-Survey و یا انجام War-Driving است. آنتن های Omni قادر به دریافت سیگنال با زاویه دید 360 درجه می باشند. آنتن های نوع Directional (مانند دیش ها و آنتن های پارابولیک) همانطور که از نام آنها پیداست زمانی کاربرد دارند که هدف شما تمرکز بر روی سیگنال های دریافتی از یک زاویه دید خاص باشد و قصد تمرکز بر روی یک هدف مشخص را داشته باشید. زاویه دید این دسته از آنتن ها محدود بوده و از 45 درجه در فاصله نزدیک و کمتر از 10 درجه در فواصل طولانی تجاوز نمی کند. همچنین دید مستقیم برای استفاده از حداکثر توان دریافتی آنتن لازم است. درمورد آنتن های Directional منظور از "فاصله نزدیک" فاصله ایی کمتر از یک کیلومتر و از "فاصله طولانی" منظور مسافتی بیش از 2-3 کیلومتر است. متذکر می شوم که هیچ یک از این اعداد و تخمین ها دقیق و علمی نبوده و محاسبه میزان دقیق این اعداد وابستگی مستقیم به شرایط جغرافیایی، سخت افزار و کیفیت ساخت تجهیزات دارد. مزیت آنتن های Directional نسبت به نوع Omni قدرت بالای دریافت سیگنال و در نتیجه برد بسیار بیشتر می باشد. ساختار آنتن های Omni میزان حساسیت آنها را نیز تحت تأثیر قرار می دهد و در قبال زاویه دید گسترده شما حساسیت را از دست خواهید داد. بهترین انوع آنتن های Omni که برای عموم قابل تهیه هستند دارای میزان Gain در حدود 15 دسی بل می باشند که در مقایسه با اعداد بین 24 تا 30 دسی بل برای آنتن های قوی از نوع Directional بسیار کمتر است. این کاربری شماست که مشخص می کند از کدام نوع آنتن استفاده کنید. در صورتی که کاربری شما از آنتن ثابت نبوده و در موارد و شرایط مختلف قصد استفاده از آنرا دارید انواع Omni با gain بین 9 تا 12 مناسب هستند. آنتن هایی که در بسته بندی کارت ها عرضه می شوند معمولاً دارای gain 5 تا 7 دسی بل هستند. در صورت نیاز به استفاده از آنتن های Directional با قیمت متعادل می توان براحتی آنتن هایی با gain بین 12 تا 17 دسی بل دست یافت که برای کار شما کاملاً مناسب هستند. هزینه برای آنتن های پارابولیک و یا دیش با gain های 20 دسی بل یا بالاتر تنها زمانی معقول است که هدف شما پوشش فواصل طولانی (بیش از 3-2 کیلومتر) باشد و علاوه بر این سخت افزار کارت شما نیز توان و قدرت کافی برای استفاده از چنین آنتنی را داشته باشد. اتصال یک آنتن 20 دسی بلی به یک کارت 100 میلی واتی تنها دامنه دریافت سیگنال شما را افزایش میدهد و کارت توان کافی برای ارسال سیگنال تا فاصله طولانی را نخواهد داشت. در بسیاری از حملات علیه شبکه های بیسیم قابلیت ارسال سیگنال به هدف نیز ضروری می باشد و در مواردی شدت سیگنال ارسالی از کارت شما ملاک موفقیت در حمله خواهد بود. به این نکته نیز توجه داشته باشید که افزایش کیفیت وgain آنتن ها از هر نوع، رابطه مستقیم و آزار دهنده ایی با قیمت آنها دارد! بطور مثال متوسط قیمت یک آنتن پارابولیک 24dBi حدوداً دو برار نمونه 15 dBi آن است، یا در مورد آنتن های Omni یک آنتن 15 dBi نزدیک به سه برابر نمونه 9 dBi آن قیمت دارد. در حال حاضر برای تهیه یک نمونه از آنتن های پارابولیک قوی در محدوده 20 dBi می بایست حداقل پنجاه هزار تومان هزینه کرد که این رقم برای آنتن های در محدوده 30 dBi به بیش از یکصد هزار تومان می رسد. پس تنها زمانی که واقعاً به آنتن های قوی نیاز دارید برای آنها هزینه کنید. انتخاب برند های معروف برای آنتن ها شما را ملزم به پرداخت هزینه های زیاد و بی جهت می کند. شرکت های داخلی ایرانی نیز محصولات مناسبی را با قیمت های کمتر از مشابه خارجی ارائه می کنند که با کمی جستجو در اینترنت می توانید بسیاری از این شرکت های ارائه دهنده تجهیزات شبکه های بیسیم را پیدا کنید.

معمولاً سوکت های مورد استفاده در آنتن های جانبی بطور مستقیم قابل اتصال به کارت های بیسیم نیستند. بدین منظور شما می بایست در زمان خرید آنتن به نوع سوکت نصب شده بر روی کابل اتصال آن و همچنین نوع سوکت اتصال آنتن بر روی کارت خود دقت کرده و در صورت نیاز اقدام به تهیه کابل های رابط و تبدیل نمایید. طول کابل های رابط نیز بر حسب نوع و کیفیت کابل استفاده شده اثر مستقیم و محسوسی بر کیفیت دریافت سیگنال خواهد داشت. هرچه میزان طول کابل رابط افزایش یابد میزان نویز و افت سیگنال نیز افزایش می یابد. با توجه به گران قیمت بودن کابل های رابط با کیفیت و همچنین در نظر گرفتن مشکل افت سیگنال که عنوان شد، می بایست سعی در کوتاه نگاه داشتن طول کابل ها نمود. بعنوان مثال افت سیگنال و میزان نویز در استفاده از کابل های با کیفیت با طول کمتر از سه متر نا محسوس است. علاوه بر خود کابل کیفیت و نحوه اتصال سوکت ها به خود کابل نیز بسیار موثر می باشند.

(سیستم عامل و درایور مناسب) : سومین آیتم نیازمندی های ما و شاید مشکل ساز ترین آنها، سیستم عامل مناسب برای کار است. بگذارید در همین ابتدا خیال شما را از این بابت راحت کنم؛ در صورتی که قصد کار بطور جدی و حرفه ایی در این زمینه را دارید سیستم عامل های مایکروسافت را فراموش کنید. بهترین گزینه ممکن سیستم عامل لینوکس می باشد و تقریباً تمام نرم افزارهای شناخته شده و ابزارهای کارامد در این زمینه نیز مبتنی بر لینوکس هستند. پس در صورتی که با این سیستم عامل آشنا نیستید بجای صرف وقت برای جستجوی مشابه ویندوزی نرم افزارها، وقت خود را صرف یاد گرفتن کار با سیستم عامل لینوکس کنید. در موارد معدودی نیز که شما قادر به استفاده از سیستم عامل ویندوز هستید، الزام استفاده از سخت افزارهای خاص و انحصاری یک شرکت/محصول در کنار نرم افزار مربوطه وجود دارد. یکی از شرکت های عرضه کننده درایورهای خاص بمنظور استفاده در ارزیابی شبکه های بیسیم از طریق سیستم عامل ویندوز شرکت WildPackets است که درایورهای این شرکت نیز دامنه بسیار محدودی از کارت ها را پشتیبانی می کند. در خانواده جدید سیستم عامل های مایکروسافت از جمله Vista و Seven تغییرات و قابلیت های جدیدی مشاهده می شود اما امکانات اضافه شده هنوز برای کارامد فرض کردن ویندوز در این شاخه کافی نیستند!

در برخی موارد خاص مانند کارت هایی که به خوبی توسط درایور های استاندارد پشتیبانی نمی شوند و یا حملات خاص، شما نیاز به Patch کردن سورس کد کرنل لینوکس را دارید. اینکه از کدام نسخه لینوکس استفاده شود اهمیت چندانی ندارد زیرا بسیاری از وابستگی های نرم افزاری مربوط به درایورهای کارت ها در سطح کرنل لینوکس است که در همه نسخه های لینوکس تقریباً یکسان است. از نگارشی استفاده کنید که فکر می کنید با آن راحت تر هستید و آزادی عمل بیشتری در مدیریت نرم افزارها به شما می دهد. Ubuntu, Debian, RedHat می توانند انتخاب های مناسبی باشند. همچنین توجه داشته باشید که کرنل سیستم عامل لینوکس را بروز کرده تا بتوانید از آخرین نسخه های درایور ها و سخت افزارهای پشتیبانی شده بهره ببرید. در صورتی که تنها دلیل استفاده شما از لینوکس ارزیابی امنیتی بوده و کاربری خاص دیگری برای آن ندارید، میتوانید از توزیع های از پیش سفارشی شده ویژه این کار استفاده کنید. بدون شک نام BackTrack را شنیده اید. بهترین و بی دردسر ترین گزینه نیز همین می باشد. اما BackTrack تنها توزیع امنیتی زنده (Live Distribution) نیست. بطور مثال Slitaz یک نمونه سفارشی شده برای استفاده بر روی نت بوک هاست(NetBook) که اکثر نرم افزارهای ذکر شده در این مقاله بصورت پیش فرض بر روی آن نصب شده و درایورهای مورد نیاز نیز در بسته سیستم عامل گنجانده شده است. در قابلیت ها و امکانات BackTrack و تنوع نرم افزارهای آن شکی نیست، اما توزیع هایی مانند Slitaz[1] با حجمی کمتر از 70 مگابایت (در مقایسه با بیش از 1 گیگابایت حجم یک ایمیج BackTrack یا نگارش های همسان آن) ممکن است شما را به فکر فرو برد. مزیت استفاده از این توزیع های آماده و زنده عدم نیاز به صرف وقت برای نصب درایور ها و نرم افزارها و درگیر شدن با مشکلات خاص مراحل این کار است. در صورت تمایل به سفارشی کردن یک سیستم عامل برای بررسی امنیتی شبکه های بیسیم نیز متن ها و راهنماهای بسیاری نوشته شده که با کمی جستجو می توان آنها را یافت.

متن راهنمای آورده شده در سایت Wirelessdefence.org[2] یکی از این نمونه هاست. این مقاله مراحل نصب نرم افزارهای مورد نیاز را پوشش نمی دهد و فرض بر این است که شما مشکلی در نصب و را اندازی نرم افزارهایی که به آنها اشاره می شود ندارید.

شنود (Sniff) شبکه های بیسیم:

در صورتی که شما پیش از این بر روی بستر شبکه های معمولی تجربه ایی در خصوص مانیتور کردن ترافیک شبکه و شنود آن داشته باشید، تمام تجربه های قبلی شما با کمی تغییر جزئی در بستر شبکه های بیسیم نیز معتبر بوده و قابل استفاده است. بحث شنود شبکه های بیسیم به خودی خود کار پیچیده و خاصی نیست اما در این بستر بدلیل عمومیت استفاده از رمزنگاری برای ارتباطات و لزوم استفاده از درایورهای خاص ظاهر کار کمی پیچیده تر بنظر میرسد. پروتکل های ارتباطی اصلی شبکه در این بستر نیز مبتنی بر تکنولوژی و پروتکل های Ethernet است با این تفاوت که کلیه این ارتباطات سطح شبکه در دل پروتکل های استاندارد 802.11 گنجانده می شود. بدین ترتیب شما علاوه بر بسته های ارسالی مربوط به فریم های Ethernet و بسته های پروتکل هایی مانند TCP, UDP, ICMP, ... فریم های کنترلی و بسته هایی مربوط به خود پروتکل 802.11 را نیز مشاهده خواهید کرد. در بستر بیسیم نیز همانند شبکه های اترنت اصل در قابلیت شنود پکت ها از لینک، امکان فعال کردن حالت Promiscuous در کارت شبکه است. در مورد کارت های بیسیم به این حالت عموماً Monitor Mode و یا RFMON گفته می شود. در سیستم عامل لینوکس بدلیل پشتیبانی کامل از امکانات بسیاری از کارت ها فعال کردن این حالت برای کارت های بیسیم به راحتی فعال کردن آن برای یک کارت شبکه معمولی می باشد اما در سیستم عامل ویندوز این کار نوعی معضل نرم افزاری حساب می شود. اینکه کارتی قابلیت فعالیت در حالت Monitor mode را دارد یا نه بجز در موارد خاصی، وابستگی مستقیم با امکانات ارائه شده توسط درایور کارت دارد. هیچ یک از تولید کننده گان کارت های بیسیم معمولی، این قابلیت را بصورت استاندارد در بسته نرم افزاری درایور (عرضه شده برای ویندوز) ارائه نمی کنند و هر آنچه در دسترس است حاصل فعالیت گروه های آزاد و کد باز است که بصورت درایور های جانبی مورد استفاده قرار می گیرد. MadWifi یک نمونه از این درایور های جانبی است. در مورد سیستم عامل ویندوز نیز مشکل دقیقاً در همین نقطه بروز می کند. تابحال شخص یا گروهی درایوری جانبی برای ارائه امکانات کامل کارت بیسیم (از جمله امکان کار در حالت monitor mode) بصورت آزاد و غیر تجاری منتشر نکرده است. در نتیجه در سیستم عامل ویندوز شما از این امکانات و قابلیت ها محروم هستید و به همین علت نیز استفاده از سیستم عامل لینوکس برای این منظور همواره توصیه شده و گاهی تنها راه است. تعدادی از شرکت های تجاری البته درایورهای اختصاصی برای استفاده در کنار محصولات نرم افزاری خاص خود تولید کرده اند و به فروش می رسانند. در اغلب موارد این درایورها تنها از سخت افزار خاصی که توسط آن شرکت تولید شده پشتیبانی کرده و برای استفاده توسط نرم افزارهای انحصاری ارائه شده بعنوان محصول تجاری شرکت قابل استفاده هستند. خلاصه همه این موارد این است که در صورت انتخاب سیستم عامل ویندوز برای بستر نرم افزارهای شنود ترافیک شبکه بیسیم، تنها راه شما استفاده از محصولات سخت افزاری و نرم افزاری تجاری وغیر آزاد می باشد. برخی از کاربران به اشتباه بسته نرم افزاری و درایور WinPcap را بدین منظور معرفی می کنند اما حتی درایور مجازی ارائه شده توسط WinPcap نیز در محیط ویندوز کارت را بصورت واقعی در حالت Promiscuous قرار نمی دهد و شما تنها قادر به دریافت پکت های مربوط به سیستم خود و یا پکت های Broadcast می باشید. علاوه بر این با توجه به اینکه WinPcap در سیستم عامل ویندوز و در بستر شبکه بیسیم، پروتکل اترنت را برای کاربر شبیه سازی می کند، شما قادر به دریافت پکت های خام کنترلی خود پروتکل 802.11 نخواهید بود و در واقع پکت ها مطابق استاندارد اترنت در اختیار شما قرار می گیرد. این امر بخصوص در زمانی که هدف شما از شنود ترافیک آنالیز فعالیت ها و مشکلات در سطح شبکه (بیسیم) باشد نا کارآمدی درایور WinPcap را نشان می دهد. درایور های تجاری ارائه شده در قالب بسته های نرم افزاری آنالیز ترافیک شبکه بیسیم تحت ویندوز این مشکل را ندارند. خود شرکت پشتیبان پروژه WinPcap البته یک کارت جانبی USB بعنوان سخت افزار ضمیمه WinPcap دارد که تنها توسط این کارت[3] سخت افزاری خاص و انحصاری شما قادر به استفاده از WinPcap در

ویندوز برای شنود ترافیک شبکه بیسیم هستید. بدین ترتیب با وجود این سخت افزار، حتی نرم افزار Wireshark نیز بخوبی در محیط ویندوز کار خواهد کرد. این سخت افزار جانبی جزو معدود بسته های نرم افزاری/سخت افزاری است که در سیستم عامل ویندوز قابلیت تولید و تزریق پکت (Packet Injection) در بستر شبکه بیسیم را نیز می دهد. اما آیا شما حاضر به پرداخت هزینه ایی بین سیصد تا پانصد دلار (بسته به مدل انتخابی) برای این بسته هستید؟ من ترجیح می دهم این مقدار پول را برای خرید یک کارت بیسیم جانبی خوب (حدود 100-150$) و چند مدل آنتن و کابل رابط (در نهایت 200$) هزینه کنم و در سیستم عامل لینوکس از قدرت بالای خروجی کارت و آنتن ها لذت ببرم. بجز محصول یاد شده از شرکت CACE، شرکت WildPackets نیز محصولی قدرتمند (از نظر قابلیت های نرم افزاری) برای سیستم عامل ویندوز ارائه می کند. محصول OmniPeek (و پیش از این نرم افزار AiroPeek) این شرکت علاوه بر خود نرم افزار یک بسته درایور[۴] اختصاصی تجاری نیز عرضه می کند که از چیپ ست های معتبری مانند Atheros و در نتیجه کارت های مبتنی بر آن پشتیبانی می کند. OmniPeek[۵] تا آنجا که من اطلاع دارم تنها محصولی است که درایور های تجاری ارائه شده در کنار آن، منحصراً بر روی سخت افزار خاص شرکت تولید کننده کار نکرده و می توان از این درایور ها برای راه اندازی سایر کارت ها و در خارج از محیط نرم افزار نیز استفاده کرد. علاقه شدید مردم در ایران به قوانین کپی رایت البته آنها را بطور قطع از تهیه و استفاده غیر قانونی از این درایورها منع میکند! با کمی جستجو می توان بسته های نرم افزاری و سخت افزاری تجاری متعددی را پیدا کرد که قابلیت هایی مشابه OmniPeek ارائه می کنند اما، محدوده قیمتی همه این محصولات غالبا بالای یک هزار دلار می باشد. باز هم هزینه! CommView-Wifi[۶] یکی دیگر از نرم افزارهای قدرتمند تحت سیستم عامل ویندوز برای تحلیل ترافیک شبکه بیسیم می باشد. لیستی از سخت افزارهای پشتیبانی[۷] شده توسط این نرم افزار توسط تولید کننده آن در دسترس است. در مورد CommView-Wifi شما این شانس را دارید که با استفاده از ابزار تست[۸] که در اختیار شما قرار داده شده از سازگاری سخت افزار کارت خود با این نرم افزار و درایور اختصاصی آن اطمینان حاصل کنید. دامنه پوشش سخت افزاری درایور عرضه شده با این نرم افزار محدود تر از بسته درایور شرکت WildPackets می باشد.

مایکروسافت در این مدت البته قدم های مثبتی را برداشته است. با عرضه نسخه های جدید (نسخه 6) توابع[۹] برنامه نویسی NDIS، قابلیت فعال کردن Monitor Mode در اختیار قرار داده شده است. این امکان در نسخه های سیستم عامل های Vista و Seven گنجانده شده است. اما یک نکته وجود دارد. NDIS تنها یک رابط برای فعال سازی و استفاده از این قابلیت است. شما همچنان به درایور یا نرم افزاری احتیاج دارید که این قابلیت را بکار گیرد. در صورتی که شما همچنان تمایل (یا اجبار!) به استفاده از سیستم عامل ویندوز را دارید مقاله ایی[۱۰] که توسط Inguardians در این خصوص نوشته شده، در کنار ابزارهای اختصاصی تولید شده توسط همین تیم، می تواند منبع مناسبی برای فراگیری روشهای استفاده ار قابلیت های ارائه شده در سیستم عامل های جدید مایکروسافت (و امکانات جدید NDIS) باشد. بخش 5.1 این مقاله توضیحاتی را در خصوص نحوه فعال کردن Monitor Mode در ویندوز ارائه می کند.

با فرض حل شدن مشکلاتی که تا اینجا در مورد سیستم عامل ویندوز مطرح شد، پس از راه اندازی موفقیت آمیز کارت و فعال کردن حالت Monitor Mode، قدم بعدی تحلیل پکت های دریافتی است که این کار معمولاً توسط نرم افزار Sniffer مورد استفاده شما بانجام می رسد. بسته به دیدگاه شما از تحلیل (رفع مشکل شبکه، و یا استخراج اطلاعاتی مانند کلمات عبور و...) نرم افزارهای مختلفی را نیز می توان استفاده کرد. اولین و بهترین انتخاب ها برای این منظور برای بسیاری از کاربران بی شک بسته های نرم افزاری Wireshark و TCPdump هستند. لازم به تذکر است که TCPDump یک Sniffer معمولی بوده و بطور کامل از 802.11 پشتیبانی نمی کند در نتیجه قادر به تحلیل پکت های مربوط به فریم های کنترلی پروتکل 802.11 نیست اما برای تحلیل ترافیک در سطح اترنت بسیار کارآمد است.

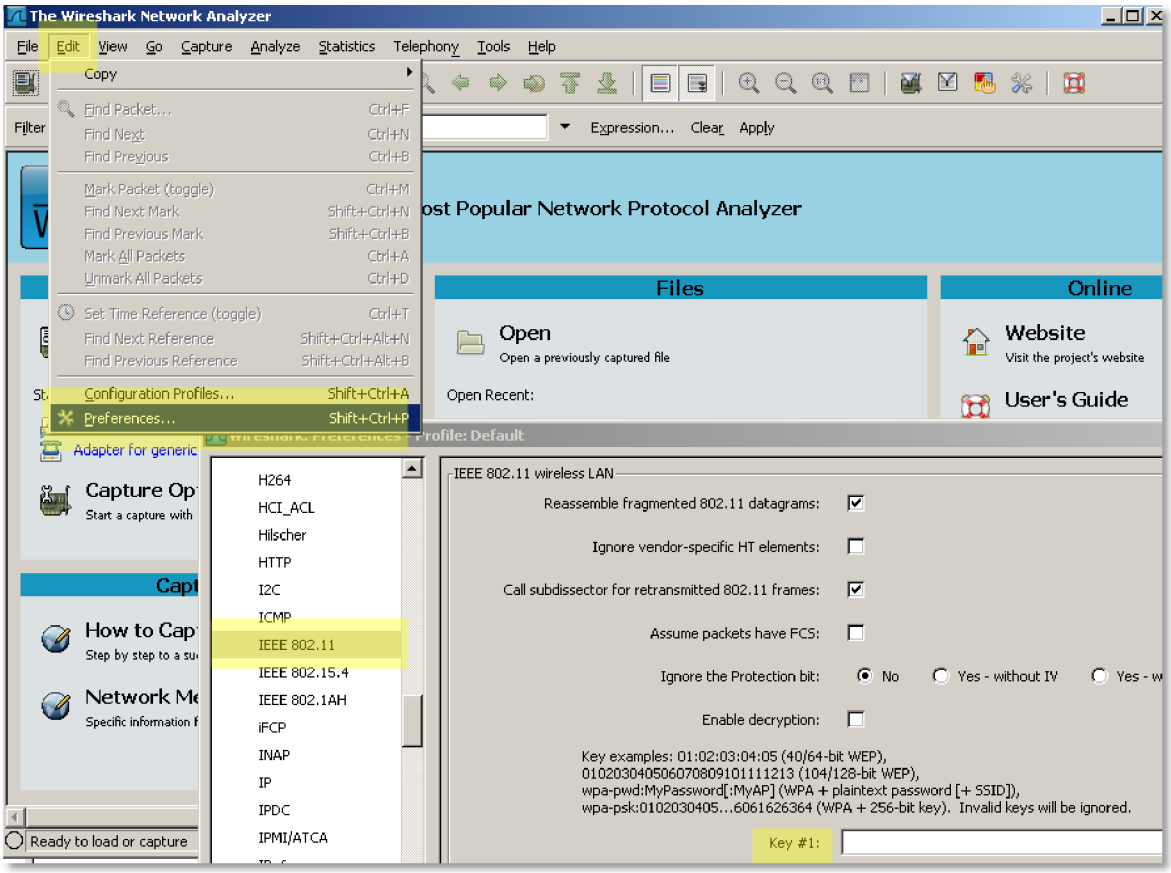
پیش از این گفته شد که مشکل اصلی در تحلیل ترافیک شبکه های بیسیم برای بسیاری از کاربران، وجود یک لایه رمزنگاری (تحت پروتکل هایی مانند WEP,WPA,WPA2 و...) حفاظت کننده از پکت هاست. در صورتی که شما قادر به شنود ترافیک یک شبکه بیسیم باشید حتی در صورت آگاهی از کلید مورد استفاده برای رمزنگاری (Pre-shared key) نرم افزارهایی مثل Dsniff, CAIN و یا پروتکل آنالایزرهایی

مانند نرم افزار Wireshark بصورت پیش فرض قادر به تحلیل ترافیک رمز شده توسط پروتکل های حفاظتی ذکر شده نمی باشند. بدین منظور و با فرض اینکه دسترسی شما به ترافیک مجاز بوده و بنا بر این از کلید رمزنگاری آگاهی دارید، اولین قدم رمزگشایی ترافیک خام است. در صورت عدم آگاهی از کلید رمزنگاری با دنبال کردن بخش های بعدی مقاله روش هایی برای بدست آوردن و استخراج این کلید معرفی خواهد شد که می توان از آنها استفاه کرد. تمامی نرم افزارهایی که از آنها یاد شد(WireShark,OmniPeek,CommView) امکان رمزگشایی ترافیک را به شما می دهند اما برای این منظور شما می بایست ابتدا کلید های مورد استفاده برای رمزنگاری را در اختیار نرم افزار قرار دهید. نحوه معرفی کلید بسته به محیط نرم افزارهای مختلف متفاوت است. با توجه به تمرکز مقاله بر استفاده از ابزارهای آزاد، نحوه انجام این کار در محیط نرم افزار Wireshark توضیح داده خواهد شد. لازم به تذکر است که نرم افزار TCPdump بصورت مستقیم و کامل از 802.11 پشتیبانی نکرده و بنا بر این امکان رمزگشایی WEP,WPA و یا WPA2 توسط آن وجود ندارد. در صورت تمایل به استفاده از نرم افزارهای تحت کنسول، می توان از Tshark (نسخه تحت کنسول Wireshark) و تنظیمات مربوط به آن استفاده کرد.

در صورتی که ترافیک توسط WEP (و در نتیجه الگوریتم رمزنگاری RC4) رمز شده باشد، آگاهی از کلید رمزنگاری در هر شرایطی کفایت می کند، یعنی شما در هر موقعیت و وضعیتی ترافیک رمز شده را شنود کنید قادر به رمزگشایی آن خواهید بود. بطور مثال در صورتی که در میان یک نشست (Session) شما به ترافیک دسترسی پیدا کنید و از آن لحظه به بعد ترافیک را شنود کنید مشکلی نخواهید داشت. در مورد پروتکل WPA و WPA2 این مورد صادق نیست و امکان رمزگشایی ترافیک ملزم به فراهم بودن شرایط خاص و چند نکته مهم اما ساده است. بر خلاف الگوریتم قدیمی WEP در الگوریتم های جدید (WPA,WPA2 و سایر الگوریتم های ارائه شده مبتنی بر روش اعتبار سنجی EAP مثل Cisco LEAP) از کلید رمز نگاری بصورت مستقیم و آنی برای رمز کردن پکت ها استفاده نمی شود. بلکه کلید رمزنگاری در واقع یکی از پارامترهای ورودی به پروسه رمزنگاری استفاده شده بوده و بر اساس این کلید رمزنگاری و ترکیب آن با پارامترهایی دیگری، داده رمزنگاری شده تولید و ارسال می شود. این امر شما را ملزم به این می کند که در صورت نیاز به شنود ترافیک بین اکسس پوینت و یک ایستگاه کاری حتماً از ابتدای برقراری ارتباط این دو، یعنی زمانی که ایستگاه کاری اصطلاحاً به اکسس پوینت متصل شده و Associate می شود، ترافیک را شنود و ذخیره کرده باشید. بدین ترتیب در صورتی که شما مانند مثال قبل در میانه یک نشست شنود خود را آغاز کنید قادر به رمز گشایی ترافیک آن نشست نخواهید بود. درک علت این موضوع مستلزم مطالعه جزیات پروتکل EAPOL[11] (EAP) و آشنایی با نحوه عملکرد این پروتکل در مرحله Handshake و رمز کردن ترافیک و اعتبار سنجی است. در چنین شرایطی شما تنها قادر به رمزگشایی ترافیک کلاینتی خواهید بود که در زمان Associate شدن آن به اکسس پوینت در حال شنود بوده اید. نکته آخر اینکه با توجه به اینکه در WPA/2 به ازای هر کلاینت یک کلید رمزنگاری جدید تولید و صادر می شود شما تنها قادر به رمزگشایی ترافیک آخرین کلاینت Associate شده در خلال شنود خود خواهید بود مگر آنکه بسته های مربوط به هر نشست بطور مجزا تحلیل و رمزگشایی شود. این محدودیت در واقع بدلیل محدودیت امکاناتی است که اکثر نرم افزارهای جاری رمزگشایی WPA/2 در اختیار قرار می دهند.

همه شرایط و مکانیزم های کاری که در مورد WPA توضیح داده شد مربوط به حالتی می باشد که از روش اعتبار سنجی Pre Shared Key استفاده شده باشد. پروتکل EAP روش های متعددی را برای اعتبار سنجی در اختیار قرار می دهد که رایج ترین آن برای استفاده در بستر بیسیم، EAP-TLS می باشد. شنود و رمزگشایی ترافیک بدست آمده از یک اکسس پوینت که در حالت Enterprise برای اعتبار سنجی تنظیم شده است (روش PSK همان حالت معمولی یا Personal است) بدلیل متفاوت بودن پروسه اعتبارسنجی و رمز کردن ترافیک، مستلزم پیاده سازی حملات پیچیده تری است که از حوصله این مقاله خارج می باشد و گذشته از این فراهم شدن شرایط حمله به WPA برای شنود در این حالت کاری نیز همیشه امکانپذیر نیست، زیرا در این حالت بخشی از اطلاعات مورد نیاز برای حمله و رمزگشایی (بعنوان مثال کلید خصوصی استفاده شده توسط کلاینت در زمان اعتبارسنجی مبتنی بر EAP-TLS) می بایست از سیستم های مسئول اعتبارسنجی در شبکه و یا کلاینت مورد نظر استخراج گردد.

بمنظور رمزگشایی ترافیک شنود شده در محیط نرم افزار Wireshark کافیست کلید های رمزنگاری مربوطه (زمانی که اکسس پوینت برای اعتبارسنجی با Pre-Shared-Key یا WEP/WPA PSK تنظیم شده باشد) که قبلاً بدست آمده به نرم افزار معرفی شود. بدین منظور از منو Edit > Preferences > Protocols > و از لیست پروتکل های موجود، IEEE 802.11 را انتخاب کرده و در بخش مربوطه کلید را وارد می کنیم. فراموش نکنید که گزینه "Enable Decryption" در این پنجره انتخاب شود. برای وارد کردن کلید فرمت های مختلفی توسط نرم افزار پشتیبانی میشود که درشرایط مختلف می توانید از هریک استفاده کنید. Wireshark تا 64 کلید رمزنگاری مختلف را بطور همزمان کنترل کرده و در صورت تعدد نشست های رمزنگاری شده و همچنین کلید های مختلف برای هر یک، بصورت خودکار کلید صحیح را انتخاب کرده و نشست را رمزگشایی می کند. در وب سایت نرم افزار نیز نحوه انجام این کار توضیح[۱۲] داده شده است.



پس از رمزگشایی شدن بسته ها شما براحتی قادر به استفاده از سایر نرم افزارهای تحلیل ترافیک و یا Sniffer های ویژه ارزیابی امنیتی (مانند CAIN, Ettercap, Dsniff) خواهید بود. در خصوص نرم افزار CAIN اگرچه خود این نرم افزار بصورت مستقیم شنود و تحلیل ترافیک شبکه بیسیم را پشتیبانی می کند اما قابلیت رمزگشایی خودکار فایل های PCAP ورودی را ندارد. اگرچه خود این نرم افزار گزینه هایی برای شکستن کلید های رمزنگاری مورد استفاده در WEP و کلمات رمز WPAرا داراست اما تنها در صورتی که خود نرم افزار نشست را شنود کرده باشد (بطور مثال با استفاده از سخت افزار انحصاری AirCap شرکت CACE) قادر به رمزگشایی ترافیک و استخراج سایر اطلاعات از آن خواهد بود. بدین منظور کافیست نشست های رمزگشایی شده مجدداً با فرمت استاندارد PCAP ذخیره شده و در سایر نرم افزارها مورد استفاده قرار گیرند. با توجه به اینکه بسیاری از نرم افزارهای تحلیل و Sniffer عمومی از WEP/WPA پشتیبانی نمی کنند، اجرای پروسه رمزگشایی و

ذخیره مجدد پکت ها بصورت رمزگشایی شده الزامیست. در محیط لینوکس نیز ابزارهایی برای رمزگشایی در دسترس هستند. ابزار AirDecap[۱۳] از مجموعه ابزارهای بسته AirCrack-NG یکی از این گزینه هاست. مکانیزم کاری این ابزار نیز مشابه موارد توضیح داده شده بوده و همان شرایط و پیش نیازها در مورد آن صدق می کند. این ابزار در محیط کنسول قابل استفاده می باشد. AirDecap پس از رمزگشایی فایل حاوی پکت ها که قبلاً ذخیره و در اختیار آن قرار داده شده (تحت فرمت استاندارد PCAP) یک فایل جدید حاوی پکت های مذکور اما بصورت رمزگشایی شده تولید می کند. استفاده از این ابزار بسیار ساده است.برای رمز گشایی یک نشست WEP کافیست آنرا بصورت زیر اجرا کنید :

```
airdecap-ng -w 11A3E229084349BC25D97E2939 wep.cap
```

دقت کنید که کلید رمزنگاری WEP می بایست بصورت هگز (Hexadecimal) به ابزار داده شود. در صورت در اختیار نداشتن مقدار هگز کافیست از یکی از ابزارهای محلی و یا آنلاین تبدیل ASCII به Hex استفاده کنید. بمنظور رمزگشایی یک نشست WPA/2 ابزار بصورت زیر مورد استفاده قرار می گیرد:

```
airdecap-ng -e 'the ssid' -p passphrase  tkip.cap
```

در مورد WPA/2 پیش از این گفته شد که کلید رمزنگاری تنها پارامتر و ورودی مورد استفاده برای پروسه و الگوریتم رمزنگاری نیست. بجز پکت های اولیه رد و بدل شده در عملیات Handshake که وجود آنها برای رمزگشایی الزامیست (تعداد آنها چهار پکت می باشد که در واقع چهار پکت اول هر ارتباط مبتنی بر WPA/2 می باشند) ، SSID یا همان نام شبکه بیسیم که ترافیک آن شنود شده نیز می بایست به ابزار داده شود. علت نیاز به آگاهی از SSID، استفاده از این عبارت بعنوان Seed در کنار کلید اصلی رمزنگاری برای تولید بسته های رمزنگاری شده است. دقت کنید که برای رمزگشایی از SSID استفاده می شود که یک رشته متنی (ASCII) شامل نام شبکه است و با BSSID که مک آدرس اکسس پوینت می باشد نباید اشتباه گرفته شود.

روش ها وابزارهای شناسایی شبکه های بیسیم:

مطالبی که پیش از این عنوان شد، مربوط به زمانی می باشد که شما قصد شنود و تحلیل ترافیک یک شبکه تحت کنترل یا شناخته شده را دارید. روش های شنود شبکه های بیسیم لزوماً با روش های کشف شبکه های بیسیم یکسان نیستند. اگرچه اساس ابزارهای کشف شبکه های بیسیم نیز همگی مبتنی بر شنود ترافیک هستند و از ابزارهای شنودی که نام برده شد برای کشف شبکه های بیسیم نیز می توان استفاده کرد، اما بهتر آن است که بمنظور افزایش دقت و سرعت روند کشف از ابزارها و روش های خاص این کار استفاده شود. بعنوان مثال استفاده از Wireshark برای صرفاً کشف شبکه های بیسیم در یک منطقه اگرچه امکانپذیر است اما انجام آن مستلزم صرف زمان زیاد برای استخراج اطلاعات از بسته های خام 802.11 شنود شده و تحلیل آنها است.بطور کلی به دو روش مختلف می توان شبکه های فعال بیسیم را شناسایی کرد.

روش اول(SSID Broadcast Discovery) تکیه بر اطلاعاتی است که هر اکسس پوینت در حالت تنظیمات پیش فرض Broadcast می کند، که همان نام شبکه یا SSID می باشد. Broadcast کردن اطلاعات اکسس پوینت در واقع با هدف سهولت اتصال و استفاده از آن صورت می گیرد و فعال بودن آن به نوعی یک ضعف امنیتی برای شبکه بیسیم حساب می شود. از جمله اطلاعات Broadcast شده می توان به نام

شبکه، مک آدرس اکسس پوینت، کانال فرکانسی مورد استفاده، نوع پروتکل، نوع مکانیزم اعتبارسنجی و رمزنگاری و اطلاعات مربوط به شدت سیگنال اشاره کرد. اکسس پوینت ها دائماً در حال ارسال این اطلاعات بنا به درخواست های کلاینت ها، در محدوده پوشش رادیویی خود هستند و یک اکسس پوینت عموماً 10 بار در ثانیه این اطلاعات را منتشر می کند. به این اطلاعات که هر اکسس پوینت بصورت خودکار دائماً در حال ارسال آنهاست اصطلاحاً Beacon گفته می شود. مشابه همین عمل در سمت کلاینت های در حال جستجو برای یک شبکه بیسیم نیز رخ می دهد. بدین ترتیب ابزارهایی که از این روش استفاده میکنند کاملاً Passive و نا محسوس عمل نمی کنند. ابزارهای مبتنی بر این روش در فاصله های زمانی بسیار کوتاه (هر از چند ثانیه) اقدام به ارسال درخواست هایی می کنند که اصطلاحاً Probe نامیده می شوند. هر اکسس پوینت در صورتی که این Probe را دریافت کند و SSID Broadcast بر روی آن غیر فعال نشده باشد، جواب probe را ارسال کرده و بدین ترتیب خود را به کلاینت در حال جستجو معرفی می کند. نرم افزار در این روش دائماً در حال ارسال probe بر روی کانال های مختلف فرکانسی و همچنین شنود Beacon های دریافتی می باشد. در نهایت نرم افزار قادر به کشف اکسس پوینت هایی که به Probe پاسخ داده اند یا Beacon ارسال کرده، و همچنین کلاینت هایی که probe را ارسال کرده اند خواهند بود.

روش دوم (802.11 Traffic Analysis) شناسایی، تحلیل ترافیک شنود شده است. در این حالت حتی اگر اکسس پوینت اطلاعاتی را Broadcast نکند و بصورت امن پیکرده بندی شده باشد باز هم اطلاعات مربوط به آن از طریق پکت های کنترلی و ارتباطی 802.11 که کلاینت های متصل به آن اکسس پوینت ارسال می کنند قابل کشف است. نکته در این روش این است که کشف چنین اکسس پوینت هایی منوط به فعال بودن آنها و ارسال یا دریافت ترافیک به آنها توسط یک کلاینت است. هرچه پکت های بیشتری به اکسس پوینت فرستاده شود و نشست های بیشتری شنود شود، اطلاعات بدست آمده نیز کامل تر خواهند بود. بطور مثال در این حالت برای کشف روش و الگوریتم رمزنگاری شنود حداقل چند پکت رمزنگاری شده الزامی می باشد و شنود پکت های کنترلی و سیگنالینگ 802.11 به تنهایی کفایت نمی کند. در مثال دیگر، در صورت غیر فعال بودن SSID Broadcast کافیست شما در زمان Associate شدن یک کلاینت به اکسس پوینت مورد نظر در حال شنود ترافیک باشید. بدین ترتیب SSID اکسس پوینت مربوطه کشف خواهد شد حتی در صورتی که از رمزنگاری در بستر شبکه استفاده شده باشد. بمنظور پوشش کلیه کانال های فرکانسی 802.11 و شناسایی شبکه های فعال در هر کانال، ابزار می بایست دائماً به کانال های مختلف سوئیچ کرده و اطلاعات آنها را تحلیل کند زیرا امکان دریافت وشنود اطلاعات از تمامی کانال های فرکانسی در یک لحظه (توسط سخت افزارهای عمومی و معمولی در دسترس) وجود ندارد.

ابزارهای متعددی در سیستم عامل های مختلف بدین منظور تولید شده اند که هر یک ویژگی ها و امکانات خاص خود را دارند، و همگی روش اول شناسایی را پشتیبانی می کنند. اما تمامی ابزارهای کشف الزاماً روش دوم شناسایی را پشتیبانی نمی کنند. نرم افزار NetStumbler در محیط ویندوز و نرم افزار Kismet در محیط لینوکس شناخته شده ترین ابزارهای کشف شبکه های بیسیم می باشند اما ده ها ابزار مشابه دیگر برای هر دو خانواده سیستم عامل وجود دارد. Kismet بی تردید کامل ترین و بهترین ابزار برای این منظور بوده و بعنوان یک استاندارد شناخته می شود. در بسیاری از مقالات و کتاب هایی که سعی در معرفی ابزارهایی برای سیستم عامل ویندوز می نمایند، NetStumbler[۱۴] تنها و بهترین گزینه موجود معرفی می شود. چندین سال از آخرین بروز رسانی این ابزار گذشته است و در این دوره تغییراتی نیز بوجود آمده. بطور مثال NetStumbler در محیط سیستم عامل های جدید مایکروسافت (Vista & Seven) به درستی عمل نمی کند. از زمان معرفی این ابزار برای سیستم عامل ویندوز، نرم افزارهای متعدد دیگری با کاربری مشابه تولید و منتشر شده اند که برخی از آنها نیز کارایی بهتری نسبت به مشابه قدیمی خود دارند. InSSIDer[۱۵] یکی از بهترین نمونه های این ابزارها می باشد که قابلیت های آن مشابه NetStumbler بوده و در محیط سیستم عامل های جدید نیز بخوبی کار می کند. این ابزار البته از توابع ارائه شده توسط سیستم عامل و درایور کارت برای شناسایی استفاده می کند که به معنی عدم توانایی در بکار گیری روش دوم شناسایی است. ابزار دیگری که در صورت کمبود امکانات سخت افزاری و نرم افزاری می توان از آن بعنوان یک حد اقل یاد کرد، Vistumbler[۱۶] می باشد که عمل کشف شبکه های بیسیم را با استفاده از

توابع خود سیستم عامل ویندوز و از طریق ابزار 'netsh' که در خود سیستم عامل های جدید مایکروسافت ارائه شده انجام می دهد و کافیست کارت شبکه بیسیم شما توسط سیستم عامل شناسایی شده باشد. از ابزارهای کارامد برای سیستم عامل ویندوز Wireless Discovery Device (Flying Squirrel)[۱۷] است که ابزار استاندارد مورد استفاده وزارت دفاع امریکا برای شناسایی شبکه های بیسیم است. این ابزار البته بصورت عمومی در دسترس نمی باشد. Eeye Wifi Scanner[۱۸] و یا AIRE Robota[19] نیز نمونه های دیگری از این ابزارها برای سیستم عامل ویندوز می باشند. یکی از ویژگی ها و امکانات جالب توجه نرم افزارهای کشف شبکه های بیسیم قابلیت آنها در استفاده از اطلاعات GPS برای ثبت موقعیت جغرافیایی شبکه های شناسایی شده است. ترکیب اطلاعات ارائه شده توسط ابزاری مثل NetStumbler یا Kismet با نقشه ها و تصاویر ماهواره ایی برای هر کسی تجربه ایی جالب و البته بسیار کاربردی است.

همانند ابزارهای شنود، ابزارهای کشف نیز بمنظور قابلیت استفاده از همه ویژگی های خود نیاز به استفاده از کارت بیسیم در حالت monitor mode دارند و در صورتی که این حالت کاری کارت در دسترس نباشد، تنها قادر به شناسایی و تحلیل اطلاعات مربوط به ایستگاه های کاری و اکسس پوینت هایی می باشند که Broadcast اطلاعات توسط آنها غیر فعال نشده باشد. پس پشتیبانی از روش دوم کشف شبکه های بیسیم در سیستم عامل ویندوز مجدداً منوط به در دسترسی بودن حالت monitor mode کارت است. همانطور که دیدید عملاً هیچ یک از ابزارهای آزاد و غیر تجاری در دسترس برای ویندوز روش دوم کشف شبکه های بیسیم را پشتیبانی نمی کنند. باز هم ویندوز ایجاد مشکل کرد!

در سیستم عامل لینوکس (و خانواده UNIX مانند OSX) مشکلات و محدودیت های ویندوز وجود ندارد. بنابر این ابزارهای تحت این سیستم عامل نیز بسیار کارامد تر و قدرتمند تر خواهند بود. Kismet[20], Airodump-NG[21], Wellenreiter[22], WifiZoo[23] همه نمونه هایی از ابزارهای برتر و شناخته شده در سیستم عامل لینوکس هستند که می توان از آنها برای کشف شبکه های بیسیم استفاده کرد، و البته تنها ابزارهای موجود نیستند. تعدد نمونه ابزارهای مشابه برای کشف شبکه های بیسیم در محیط لینوکس بسیار بیشتر از ویندوز است برخی از آنها از ابزارهای یاد شده بعنوان هسته کاری خود استفاده کرده و قابلیت یا ویژگی را به نرم افزار اصلی اضافه کرده اند. تمرکز این مقاله بر استفاده از Kismet وAiroDump برای کشف و شناسایی می باشد.

حتی در سیستم عامل لینوکس نیز همه ابزارهای کشف ذکر شده بصورت پیش فرض و بدون تنظیمات و فراهم بودن پیش نیاز ها قادر به ارائه سرویس و عرضه قابلیت های خود نیستند. در سیستم عامل لینوکس نیز لزوم استفاده از کارت در حالت monitor mode وجود دارد منتها مشکل تجاری بودن درایور ها و محدود بودن دامنه سخت افزاری پشتیبانی شده توسط این درایورها در لینوکس وجود ندارد. پروژه MadWifi[۲۴] که نام آن در بخش های قبلی مقاله آورده شد یکی از بهترین و کامل ترین پروژه های کد باز درایور کارت های بیسیم در محیط سیستم عامل لینوکس می باشد. قابلیت ها و پشتیبانی گسترده از این پروژه سبب شده تا بسیاری از ابزارهای بررسی و شناسایی شبکه بیسیم، آنرا بعنوان یک درایور استاندارد و پیش فرض حساب کرده و از آن استفاده کنند. یکی از قابلیت های بسیار جالب که این درایور در اختیار قرار می دهد، تعریف چندین اینترفیس شبکه مجازی مبتنی بر یک کارت سخت افزاری می باشد. بدین ترتیب شما می توانید همزمان از چند اینترفیس با تنظیمات متفاوت استفاده کنید. مثلاً یک اینترفیس برای اتصال به اینترنت از طریق بیسیم، و اینترفیس دومی برای شنود پکت ها که در حالت monitor mode قرار دارد. در شرایط و درایور های معمولی یک کارت نمی تواند بطور همزمان هم در حالت Promiscuous قرار داشته و هم به شما امکان برقراری ارتباط با اکسس پوینت و شبکه را بدهد. قابلیتی مشابه آنچه در حال حاضر در Madwifi-ng پیاده سازی شده در سطح کرنل در ویندوز Seven نیز گنجانده شده است (تحت نام VirtualWiFi[۲۵]) اما بدلیل اینکه هنوز تولید کننده ها درایوری برای آن عرضه نکرده اند این قابلیت غیر فعال و در عمل غیر قابل استفاده است. البته تکنولوژی پیاده سازی شده در ویندوز با دید دیگری عرضه شده و هدف آن قابلیا اتصال همزمان به دو اکسس پوینت (تبدیل یک کارت سخت افزاری به دو کارت مجازی و مستقل از هم) بوده است. بسته های درایور دیگری نیز برای لینوکس در دسترس هستند که در حال حاضر اغلب آنها در بسته کرنل لینوکس گنجانده شده

اند و برای استفاده از آنها کافیست نسخه کرنل مورد استفاده در بر گیرنده آنها باشد. از این دسته درایور ها میتوان به درایور های مربوط به کارت های Intel اشاره کرد که با نام IPW (مثلاً ipw3945) شناخته می شوند، و یا درایور استاندارد کارت های بیسیم مبتنی بر چیپ ست های Atheros که درایور آنها با نام Ath5k و یا Ath9k شناخته می شود. همه درایورهای یاد شده قابلیت فعال کردن حالت Monitor mode را بصورت یک قابلیت استاندارد در اختیار قرار می دهند، درست بر خلاف درایور های استاندارد ویندوز که هیچ یک این قابلیت را بصورت استاندارد عرضه نمی کنند، اما قابلیت Packet Injection در درایورها مجدداً نیاز به بررسی و گاهاً patch کردن سورس کد درایورها یا کد کرنل لینوکس را دارد.

در صورتی که کارت شما بصورت پیش فرض توسط سیستم عامل شناسایی نشده است (اما اطمینان دارید که جزو سخت افزارهای سازگار است) ممکن است ماژول کرنل مربوط به درایور کارت مورد نظر بصورت خودکار توسط سیستم بارگذاری نشده باشد. در چنین شرایطی میتوانید درایور مربوطه را توسط دستور 'modprobe' بارگذاری کنید. بطور مثال برای آگاهی از لیست درایور های موجود و فعال شده در کرنل برای سری کارت های Intel میتوانید از دستور 'modprobe –l | grep ipw' استفاده کنید و یا در صورت وجود درایور مربوط به کارت شما در سیستم اما بارگذاری نشدن آن بصورت خودکار، آن را توسط دستور 'modprobe ipwraw' بارگذاری کنید. این دستور درایور خام مربوط کارت های Intel را فعال می کند. در مورد برخی از سری های کارت های اینتل درایوری که کرنل بصورت پیش فرض برای راه اندازی کارت استفاده می کند با ابزارهایی مانند Kismet یا ابزارهای بسته Aircrack-NG همخوانی ندارد. در چنین شرایطی این درایور می بایست ابتدا غیر فعال شده و سپس کارت با درایور مربوطه مجدداً راه اندازی گردد. بطور مثال سری3940 کارت های اینتل دارای این مشکل می باشند و با درایور پیش فرض ipw3945 راه اندازی می شوند. برای غیر فعال کردن این درایور از دستور 'modprobe –r ipw3945' استفاده شده و پس از آن درایور سازگار با دستور 'modprobe ipwraw' راه اندازی می شود. جزئیات بیشتری در این خصوص و برای مواردی که نیاز به استفاده از یک درایور خاص بوده و یا کارت شما به درستی توسط سیستم شناسایی نمی شود در وب سایت Wiki[۲۶] و فروم[۲۷] مربوط به BackTrack آورده شده است.

یک راه دیگر نیز برای مدیریت درایورهای مربوط به کارت های شبکه بیسیم در سیستم عامل لینوکس و جود دارد و آن هم استفاده از اسکریپت ارائه شده در بسته نرم افزاری Aircrack-ng می باشد. اسکریپت Airdriver-ng به شما این امکان را می دهد که براحتی لیستی از ماژول ها و درایورهای کرنل مربوطه را مشاهده کرده، آنها را نصب و بارگذاری کنید و یا آنها را غیر فعال کنید. برای بسیاری از افراد که آشنایی کافی با جزئیات درایورها و نحوه مدیریت آنها ندارند، این اسکریپت می تواند بسیار کارآمد و مفید باشد. پیشنهاد می شود حتماً توضیحات مربوط به این ابزار و نحوه استفاده از آن را در وب سایت نرم افزار مطالعه کنید.

```
unknown aircrack-ng # airdriver-ng
Found kernel: 2.6.21.5
usage: airdriver-ng <command> [drivernumber]
        valid commands:
                supported               - lists all supported drivers
                kernel                  - lists all in-kernel drivers
                installed               - lists all installed drivers
                loaded                  - lists all loaded drivers
                ------------------------------------------------------
                insert <drivernum>      - inserts a driver
                load <drivernum>        - loads a driver
                unload <drivernum>      - unloads a driver
                reload <drivernum>      - reloads a driver
                ------------------------------------------------------
                compile <drivernum>     - compiles a driver
                install <drivernum>     - installs a driver
                remove <drivernum>      - removes a driver
                ------------------------------------------------------
                compile_stack <stacknum>        - compiles a stack
                install_stack <stacknum>        - installs a stack
                remove_stack <stacknum> - removes a stack
                ------------------------------------------------------
                install_firmware <drivernum>    - installs the firmware
                remove_firmware <drivernum>     - removes the firmware
                ------------------------------------------------------
                details <drivernum>     - prints driver details
                detect                  - detects wireless cards
```

بررسی نرم افزار Kismet : شاید عنوان کردن قابلیت های این ابزار در قالب بخش خلاصه ایی از یک مقاله صحیح نباشد زیرا معرفی کامل این نرم افزار و آشنایی با تمام قابلیت ها و جزئیات آن خود نیازمند نگارش یک متن مفصل است، که البته قبلاً تحت عنوان مستندات[۲۸] این پروژه تا حد زیادی بانجام رسیده است. اساس کار Kismet بصورت کلاینت سروری می باشد. بدین معنی که هسته اصلی نرم افزار که وظیفه تحلیل و ثبت اطلاعات را دارد در پس زمینه اجرا شده و بخش رابط کاربری نرم افزار بصورت مجزا میتواند اجرا شده و با هسته یا همان Daemon نرم افزار Kismet ارتباط برقرار کند. سرور و کلاینت می توانند بر روی دو سیستم کاملاً مجزا و در محیط شبکه هم اجرا شوند و با یکدیگر ارتباط برقرار کنند اما عموماً هر دو بخش بر روی یک سیستم مورد استفاده قرار می گیرند. این ویژگی سبب شده تا پروژه های متعددی با تمرکز بر روی افزایش کیفیت و کارایی رابط کاربری Kismet بوجود آید، بدون اینکه تولید کننده رابط کاربری خود را درگیر پیچیدگی های سرور یا هسته نرم افزار نماید. Q-Kismet[۲۹] یک نمونه جالب از این پروژه ها می باشد که البته تمامی قابلیت هایی که رابط کاربری استاندارد Kismet ارائه می دهد در آن پیاده سازی نشده است.

پروژه Kismet در حال حاضر دارای دو شاخه کد اصلی می باشد و شاخه جدید تر که در واقع پیاده سازی جدید و کاملاً بازنویسی شده کد قدیمی می باشد تحت نام Kismet New-core در دسترس است که البته هنوز تحت تولید بصورت فعال بوده و بصورت رسمی تحت یک نسخه پایدار (Stable release) منتشر نشده است. پشتیبانی از Kismet سنتی کماکان بصورت زنده ادامه دارد و بروز رسانی می شود.

در صورتی که در محیط Backtrack(4) قصد بررسی این نرم افزار را دارید یک اسکریپت wrapper عمل اجرای سرور و کلاینت را برای شما انجام خواهد داد. کافیست در محیط کنسول سیستم عامل، دستور 'kismet' را اجرا کنید تا به رابط کاربری و در پس زمینه آن سرویس Kismet دست یابید. در حالت پیش فرض بر روی نوت بوک ها این اسکریپت ممکن است به درستی عمل نکرده و سرویس پس زمینه اجرا نگردد. علت آن نامگذاری های متفاوت درایورها بر روی اینترفیس های شبکه بیسیم است. بسته به نوع کارت شما ممکن است این نام متفاوت باشد و نام اینترفیس مورد نظر با نامی که بصورت پیش فرض در فایل پیکره بندی Kismet آورده شده، همخوانی نداشته باشد. فایل پیکره

بندی از مسیر ‘/usr/local/etc/kismet.conf’ قابل دسترسی می باشد و مرجع بخش سرور نرم افزار برای دریافت تنظیمات است. در صورت بروز مشکل در اجرای این اسکریپ wrapper می بایست یا فایل پیکره بندی را ویرایش کرده و یا سرور Kismet را با پارامترهای لازم از خط فرمان فراخوانی کنیم. در Kismet نام و نوع اینترفیس ورودی برای دریافت و شنود پکت ها تحت عنوان ‘Capture Source’ شناخته می شود. جزئیات کامل مربوط به پیکره بندی Capture source در مستندات نرم افزار در بخشی به همین نام آورده شده است، اما بطور خلاصه Kismet برای استفاده از یک اینترفیس باید نام آن و همچنین نوع درایور راه انداز آن را بداند. نام اینترفیس مشخص بوده و با اجرای دستور ‘ifconfig’ می توان از آن آگاه شد. در کارت های مبتنی بر چیپ Atheros نام گذاری معمولا بصورت ... ,ath0, ath1 می باشد. در صورت استفاده از درایور Madwifi نام گذاری بسته به نوع کارت ممکن است بصورت ...wlan0, wlan1, باشد. نوع درایور مورد استفاده و نام اختصاری آن بر طبق نامگذاری های Kismet را نیز می توانید در مستندات نرم افزار مشاهده کنید. پس از مشخص شدن این دو مورد می توان به دلخواه فایل پیکره بندی را تغییر داده (روش پیشنهادی) یا هر بار سرور نرم افزار را از طریق خط فرمان با پارامترهای لازم فراخوانی کرد. در فایل پیکره بندی این تنظیمات در قسمت ‘source=’ معرفی می شوند. برای آگاهی از تنظیمات جاری فایل پیکره بندی کافیست دستور زیر را اجرا کنید :

```
unknown ~ # cat /usr/local/etc/kismet.conf |grep source=
# source=sourcetype,interface,name[,initialchannel]
source=none,none,addme
```

همانطور که مشاهده میکنید ترتیب و نحوه معرفی نیز آورده شده است. منظور از sourcetape نام اختصاری (Alias) است که Kismet از آن برای معرفی نوع درایور استفاده می کند، مثلا madwifi. Interface نامی است که درایور به سخت افزار شما اختصاص داده است، مثلاً wlan0 و در نهایت name عنوانی است که در رابط کاربری نرم افزار، برای نشان دادن نام اینترفیس به شما از آن استفاده می شود. با ویرایش این خط از فایل پیکره بندی، در صورت صحیح بودن سایر تنظیمات در فایل میتوان سرور Kismet را بدون پارامتر خاصی از طریق خط فرمان و دستور ‘kismet_server’ اجرا کرد. در صورت تمایل به اجرای سرور توسط پارامترهای مربوط به Capture source اجرای kismet_server بصورت زیر و مطابق روش تعریف در فایل پیکرده بندی خواهد بود. دقت کنید که سوئیچ c– با حروف کوچک تایپ شده است. همچنین در صورت تمایل می توانید با اضافه کردن سوئیچ daemonize– برای سرور مشخص کنید که پس از اجرا کار خود را در پس زمینه ادامه دهد و دیگر در محیط کنسول گزارشی ارئه ندهد.

```
kismet_server –c driver type,interface name, custom name
unknown ~ # kismet_server -c madwifi,wlan0,card1
```

پس از اجرای سرور و یا اجرای آن بصورت daemonized شما کافیست رابط کاربری را با دستور ‘kismet_client’ اجرا کنید تا به خروجی های سرور Kismet دست یابید. دقت کنید که سرور یا کلاینت Kismet برای ادامه کار خود نیازی به دسترسی root ندارند و اجرای Kismet نیز با سطح دسترسی root از نظر امنیتی توصیه نمی شود. نکته دیگر اینکه در شرایط معمولی و در صورت عدم وجود مشکل در درایور و یا سازگار نبودن کارت و درایور با حالت Promiscuous که در لینوکس اصطلاحاً Monitor mode خوانده می شود، خود Kismet بصورت خودکار کارت را در حالت Monitor mode قرار خواهد داد. در صورتی که از Madwifi استفاده می شود می توان یک اینترفیس مجازی جدید تعریف کرده، آنرا در حالت Monitor mode قرار داده و به Kismet معرفی کرد. در صورت نیاز به فعال کردن این حالت برای کارت بصورت دستی، می توان از ابزار airmon-ng که جزو بسته Aircrack-NG است استفاده کرد. در بخش بعدی مقاله در این مورد توضیحاتی ارائه خواهد شد.

Kismet قابلیت بسیار جالبی دارد و آن امکان معرفی و استفاده همزمان از چند اینترفیس برای شنود است. این قابلیت بخصوص در زمان War-Driving که قصد شما مانیتور کردن چندین کانال فرکانسی و دامنه باند است کاربرد پیدا می کند. قبلاً گفته شد که کارت در آن واحد تنها قادر به مانیتور کردن یک باند و محدوده فرکانسی می باشد. بنا بر این در زمانی که کارت شما در حال شنود کانال 6 محدوده فرکانسی باند G است، اگر سیگنالی از یک کانال دیگر حتی بر روی همین باند وجود داشته باشد کارت شما و در نتیجه Kismet آنرا دریافت نخواهد کرد. این مورد بخصوص در زمان متحرک بودن سیستم شنود (war-driving) و با توجه به اینکه شما تنها چند ثانیه ممکن است در محدوده پوشش رادیویی اکسس پوینت هایی که در حال عبور از نزدیکی آنها هستید قرار داشته باشید، نمود پیدا می کند و منجر به از دست دادن و ثبت نشدن اطلاعات برخی از اکسس پوینت ها شود. در زمان war-driving من سعی میکنم حداقل از دو اینترفیس و کارت بصورت همزمان استفاده کنم. یکی کارت داخلی خود نوت بوک و دیگری کارت(های) جانبی اضافه شده به سیستم. در صورتی که سیستم مورد استفاده برای شنود و کشف تحرک چندانی ندارد یا سرعت تحرک بسیار کم است (مانند تحرک بدون وسیله نقلیه) نیازی به استفاده از چند کارت بصورت همزمان وجود ندارد.

بررسی بسته نرم افزارهای Aircrack-NG : مجموعه ابزارهای ارائه شده در بسته Aircrack-NG شاید کامل ترین بسته ابزار برای پیاده سازی حملات بر روی شبکه های بیسیم بوده و برخی از ابزارهای این مجموعه نیز از نظر کارایی منحصر بفرد می باشند. این مقاله پوشش دهنده و معرف کلیه ابزارهای بسته نبوده و در هر بخش تنها ابزاری که متناسب با آن است مورد بررسی قرار می گیرد. لیست[۳۰] کامل ابزارهای این بسته بهمراه توضیحاتی در مورد هر ابزار و کاربرد آن در وب سایت Aircrack-ng آورده شده است. همچنین Wiki کامل این بسته نرم افزاری توضیحات بسیار کامل و جامعی را از مراحل اولیه نصب تا کاربری پیشرفته ابزارها در اختیار قرار می دهد.

دراین بخش از مقاله دو ابزار Airmon-NG و AiroDump-NG مورد بررسی قرار می گیرند. آوردن نام ابزار Airodump-NG در لیست معرفی ابزارهای شناخته شده و کار آمد برای کشف شبکه های بیسیم، شاید طبقه بندی درستی نباشد و بسیاری این ابزار را در کنار ابزار اصلی aircrack-ng بعنوان بسته حمله به WEP/WPA بشناسند. اما در واقع Airodump-NG در کنار Kismet جزو معدود ابزارهایی هستند که بطور عینی قابلیت تشخیص و کشف شبکه های وایرلس به روش کاملاً غیر فعالانه (Passive) و از طریق تحلیل پکت های دریافتی را دارند. در صورتی که هدف شما از استفاده از Airodump تنها کشف شبکه های وایرلس باشد کافیست ابتدا کارت را در حالت Monitor Mode قرار داده و سپس ابزار را اجرا کنید. بمنظور کنترل وضعیت کارت ابزار Airmon-NG در بسته Aircrack وجود دارد که به سادگی امکان تغییر وضعیت کارت و ساختن اینترفیس های جدید (مطابق قابلیت درایور MadWifi که پیش از این عنوان شد) را به شما می دهد. با فرض اینکه اینترفیس کارت شبکه بیسیم شما wlan0 است، برای فعال کردن حالت Monitor mode کافی است دستور 'airmon-ng start wlan0' را اجرا کنید. پس از اجرا گزارشی از وضعیت کارت نیز به شما نشان داده خواهد شد. برای غیر فعال کردن حالت monitor mode کافیست دستور 'airmon-ng stop wlan0' اجرا گردد. در شرایط استفاده از madwifi در صورتی که شمل اقدام به فعال کردن این حالت کنید بجای تغییر حالت اینترفیس اصلی، یک اینترفیس مجازی که در حالت monitor قرار دارد با نام جدید ساخته خواهد شد که از این اینترفیس بجای wlan0 باید استفاده کنید. برای حذف اینترفیس نیز کافیست دستور 'airmon-ng stop {new interface name}' را اجرا کنید. مثلاً اگر اینترفیس جدید با نام ath0 ایجاد شده باشد؛ 'airmon-ng stop ath0'. حال می توانید ابزار Airodump-NG را فعال کرده و خروجی های آن را مشاهده کنید. کافیست دستور 'airodump-ng ath0' را اجرا کنید. ابزار بصورت خودکار شنود را آغاز کرده و دائماً به کانال های مختلف سوئیچ می کند. در زمان اجرا بخش پایین گزارش مربوط به لیست کلاینت های کشف شده و Probe های ارسالی آنهاست و اینکه هر کلاینت به کدام اکسس پوینت Associate شده. در صورتی که قصد شنود ترافیک بر روی باند خاصی (A/B/G) و یا کانال خاصی را دارید و احساس می کنید سوئیچ کردن بین کانال ها موجب از دست رفتن پکت هایی می شود که به آنها نیاز دارید، توسط پارامترهای اضافه می توانید Airmon-ng را تنظیم کنید. مثلاً در صورتی که قصد شما فقط بررسی باند A باشد

دستور اجرا 'airodump-ng ath0 –band a' خواهد بود. بدین ترتیب ابزار فقط بین کانال های فرکانسی مربوط به باند 802.11a سوئیچ خواهد کرد. در صورتی که قصد شما شنود بر روی یک یا چند کانال خاص باشد دستور اجرای ابزار 'airodump-ng ath0 –channel 6,11' خواهد بود. بدین ترتیب ابزار فقط بین دو کانال 6 و 11 سوئیچ کرده در نتیجه امکان از دست دادن یک probe یا پکت کاهش می یابد. بصورت پیش فرض Airodump-ng تنها کانال های مربوط به فرکانس 2.4 Ghz را مانیتور می کند (802.11b/g/n) که در صورت نیاز می توان آن را محدود تر یا بیشتر کرد. توضیحات کامل و جامع مربوط به این ابزار در صفحه Wiki مربوط[۳۱] به آن آورده شده است. توصیه همواره بر استفاده از Kismet می باشد اما در شرایطی که قصد شما آگاهی سریع و بی دردسر از محیط اطراف در زمان Surveyباشد و یا تنها قصد بدست آوردن اطلاعاتی مانند SSID,BSSID, Client MAC و شدت سیگنال را پیش از شروع یک حمله دارید، فکر نمی کنم ابزاری بی دردسر تر و سریع تر از Airodump وجود داشته باشد. حتی دستور 'iwconfig' لینوکس نیز برای این کاربرد خروجی آشفته ایی دارد. Airodump-NG دارای یک فرمت خروجی اختصاصی بصورت فایل XML یا CSV در کنار فایل PCAP است، همانند آنچه در Kismet وجود دارد. علاوه بر آنچه در زمان اجرای ابزار مشاهده می شود، کلیه اطلاعات بدست آمده از شبکه ها و اکسس پوینت ها در این فایل استاندارد نیز ذخیره می شود. این خروجی به فایل بصورت پیش فرض فعال نبوده و برای استفاده از آن می بایست از سوئیچ beacons– استفاده شود. آگاهی از اطلاعاتی که Airodump یا Kismet در مورد یک شبکه در اختیار قرار می دهند برای پیاده سازی هر نوع حمله به شبکه هدف ضروری می باشد.

Packet Injection Test: پیش از آشنایی با جزئیات حملات و نحوه انجام آنها ذکر یک نکته دیگر الزامیست. قبل از انجام یک حمله موفقیت آمیز لازم است ما از کارکرد صحیح سخت افزارها (آنتن، کارت شبکه) و نرم افزارهای مورد استفاده خود (درایور نصب شده، کارکرد سیستم عامل) برای حمله اطمینان حاصل کنیم. یک اشکال شایع که بسیاری از کاربران را به اشتباه می اندازد این فرض غلط است که چون کارت شبکه بیسیم آنها در لینوکس (یا ویندوز) شناسایی شده و امکان اتصال به شبکه بیسیم توسط آن بررسی و تآیید شده، بنا بر این برای حمله نیز آماده است. درایوری که به شما اجازه اتصال به شبکه می دهد ممکن است لزوماً به شما امکان انجام Packet Injection را ندهد. در بخش های قبلی روش ها و ابزارهایی برای راه اندازی کارت شبکه بیسیم در محیط ویندوز معرفی شد که حتی قابلیت Monitor Mode را نیز در اختیار قرار می دهند. اما هیچ یک از این روش ها (بجز موارد خاص تجاری) هنوز امکان تولید و تزریق پکت های خام را در اختیار قرار نمی هدند. جدای از این موضوع ممکن است قدرت سیگنال دریافتی و ارسالی نیز کافی نباشد، یعنی علی رغم قابلیت دریافت سیگنال توسط کارت شما (بدلیل قوی بودن منبع ارسال) ممکن است سیگنال های ارسالی از کارت شما آنقدر قوی نباشد که به اکسس پوینت برسد. همانطور که گفته شد قابلیت ارسال پکت در انجام بسیاری از حملات به شبکه های بیسیم الزامی است. برای اطمینان از هر دو مورد ذکر شده لازم است قبل از انجام حمله به یک شبکه، این موارد بخصوص قابلیت دریافت سیگنال های ارسالی ما توسط اکسس پوینت کنترل شود. ابزار Aireplay-ng در بسته Aircrack با اضافه کردن قابلیت تست به نرم افزار کار بررسی این موارد را بسیار آسان کرده است.

انجام تست توسط سوئیچ "9-" و یا "test--" فعال می شود. در حالت پیش فرض اجرا و در صورتی که Aireplay-ng بدون هیچ تنظیم خاصی برای تست اجرا شود("aireplay-ng -9 wlan0")، ابزار به جستجوی اکسس پوینت های فعال بر روی کانال های مختلف پرداخته و بر روی هر اکسس پوینت تست هایی را انجام می دهد. در صورت نیاز می توان ابزار را تنظیم کرد تا تست را تنها برای یک اکسس پوینت خاص بانجام رساند. نحوه فراخوانی ابزار برای این کار بدین صورت است :

```
aireplay-ng -9 -e AP-name -a 00:de:ad:ca:fe:00 ath0
```

نحوه انجام بررسی توسط ابزار بدین صورت است که پس از جستجو و شناسایی اکسس پوینت ها از طریق دریافت Beacon های ارسالی از آن که دائماً در حال انجام است و یا دریافت پاسخ از اکسس پوینت که به Broadcast beacon های کلاینت داده می شود، 30 بار به اکسس پوینت Probe فرستاده می شود. Probe های ارسالی از اکسس پوینت تقاضای ارائه اطلاعاتی درباره خود می کنند. با ارسال یک Probe توسط ابزار اولاً قابلیت ارسال (Packet Injection) کنترل می شود، ثانیاً در صورت دریافت جواب از اکسس پوینت می توان مطمئن شد که سیگنال ارسالی از سمت ما آنقدر قوی بوده است که به اکسس پوینت رسیده و او را وادار به ارسال پاسخ کند. دلیل تکرار این عمل نیز تخمین کیفیت ارتباط بر اساس تعداد درخواست های رسیده و پاسخ داده شده است. بنابر این ابزار از هر دو روش کشف و شناسایی

اکسس پوینت ها که پیش از این مورد بحث قرار گرفت استفاده می کند. قابلیت مشخص کردن نام و BSSID اکسس پوینت امکان بررسی و تست اکسس پوینت های مخفی شده که SSID Broadcast بر روی آنها غیر فعال شده را نیز به ما می دهد. جزئیات کامل در خصوص نحوه انجام تست و حالت های مختلف تست توسط Aireplay-ng –test در صفحه wiki مربوط به ابزار[۳۲] در دسترس است.

با آشنا شدن با روش ها و چند ابزار برای کشف و شناسایی شبکه های بیسیم در یک محدوده، حال میتوان اقدام به بررسی دقیق تر هر یک از شبکه های کشف شده و بررسی نوع حملات امکانپذیر برای هر یک نمود. چند مورد از رایج ترین حملات علیه شبکه های بیسیم مورد بررسی قرار گرفته و ابزارهایی نیز برای پیاده سازی این حملات معرفی خواهد شد.

بررسی حملات علیه پروتکل های امنیتی شبکه های بیسیم

حملات سنتی به WEP :

Wired Equivalent Privacy[۳۳] یا به اختصار WEP، یکی از پروتکل های امنیتی قدیمی می باشد که سالها پیش (1997) بمنظور رفع برخی از ضعف های جدی امنیتی پروتکل IEEE 802.11 ارائه شده و بسرعت مورد استقبال قرار گرفت. استفاده از WEP ساده ترین راه محافظت از پروتکل 802.11 برای جلوگیری از حملات شنود بود زیرا خود پروتکل 802.11 بصورت استاندارد پکت ها را اصطلاحاً Clear-text منتقل کرده و منتشر می کند. بنا بر این هر کسی که در محدوده پوشش رادیویی قرار داشته باشد براحتی قادر به شنود کلیه ترافیک رد و بدل شده از طریق بستر شبکه بیسیم خواهد بود، درست مثل اینکه سیستم های متصل به اکسس پوینت از طریق یک هاب به هم مرتبط شده باشند و هر کسی بدون نیاز به دسترسی فیزیکی نیز بتواند به آن هاب متصل شده و اطلاعات را شنود کند. با گذشت چند سال از بکار گیری WEP متخصصین رمزنگاری تحقیقات علمی و عملی مختلفی را بر روی مکانیزم و الگوریتم رمزنگاری مورد استفاده در WEP انجام دادند که منجر به شناسایی ضعف های امنیتی جدی برو روی الگوریتم رمزنگاری WEP شد. شروع انتشار عمومی نتایج تحقیقات مربوط به سال 2001 بود که جرقه بسیاری از حملات practical به این پروتکل نیز در همان زمان زده شد. یکی از جامع ترین تحقیقات انجام شده در مورد مشکلات WEP مقاله ایی[۳۴] بود که در سال 2001 در IEEE منتشر شده و چندین حمله تئوری ممکن به RC4[۳۵] را تشریح کرده و مورد بررسی قرار داد. مطالب اولیه و تئوری های مطرح شده در این مقاله نقطه شروع و شکل گیری بسیاری از حملات پیشرفته بود که در سالهای بعد منتشر شد. در حال حاضر تقریباً 16 روش و تکنیک مختلف حمله به WEP بصورت تئوری مطرح شده و شناخته شده است. بیش از نیمی از این حملات هنوز بصورت عملی و Practical قابل انجام نبوده و یا ابزاری برای پیاده سازی این حملات بصورت عمومی منتشر نشده است. اغلب ابزارهایی نیز که برای حمله به WEP در سال های بین 2001 تا 2006 منتشر شد در واقع پیاده سازی عملی همان مفاهیمی بوده است که تحقیقات آن در سال 2001 و پس از آن بصورت تئوری بانجام رسیده بود. قلب پروتکل WEP و هسته اصلی رمزنگاری اطلاعات در آن مبتنی بر الگوریتم رمزنگاری RC4 می باشد و تمرکز بسیاری از حملات تئوریک و عملی شده علیه WEP نیز مبتنی بر ضعف های امنیتی کشف شده در همین الگوریتم است. اساس کار همه حملات سنتی علیه WEP در یک نقطه مشترک می باشد و آن هم تکیه بر آنالیز پکت های رمز شده، شناسایی نقاط تکرار (Collision) در الگوریتم و داده های رمز شده و جمع آوری تعداد زیادی از این پکت های خاص برای تحلیل نهایی است که در نهایت منجر به کشف کلید مورد استفاده برای رمزنگاری می گردد. درک جزئیات این حملات تئوری و عملی نیازمند آگاهی از نحوه کارکرد WEP، و شناخت الگوریتم RC4 می باشد. در عین حال بدلیل وجود ابزارهای متعدد که این حملات را بصورت عملی پیاده سازی کرده اند، استفاده عملی از این حملات برای شخصی که آگاهی از جزئیات تئوری کار ندارد نیز میسر شده است. همانطور که ذکر شد ریشه و اساس همه حملات تحلیل بسته های رمز شده به روش های گوناگون و سعی در استخراج کلید رمزنگاری (WEP Key) است. تکنیک و ابزارهای های مختلفی که برای پیاده سازی وجود دارند همگی در واقع روش هایی هستند برای بدست آوردن تعداد

بیشتری پکت رمز شده در کوتاه ترین زمان ممکن و همچنین شرایط کاری مختلف. نقطه پایانی کار همه این ابزارها و تکنیک ها به ضعف الگوریتم RC4 و شکستن آن ختم می شود.

مکانیزم کاری WEP : اگر بخواهیم به زبان بسیار ساده مکانیزم کاری WEP و رمزنگاری در آن را تشریح کنیم می توان آنرا اینطور بیان کرد؛ WEP برای حفظ محرمانگی (Confidentiality) داده های انتقالی از الگوریتم رمزنگاری RC4 و برای کنترل صحت داده ها (Integrity) از روش کنترلی CRC-32[۳۶] استفاده می کند. WEP استاندارد برای رمزنگاری از یک کلید رمزنگاری 40 بیتی استفاده می کند. با توجه به اینکه RC4 یک الگوریتم رمزنگاری خطی (Stream Cipher[۳۷]) می باشد، بجز خود کلید رمزنگاری به یک رشته ورودی تصادفی دیگر نیز برای انجام فرایند رمزنگاری نیاز است. در WEP این رشته تولید شده که 24 بیت طول دارد به کلید اصلی اضافه شده و چرخه رمزنگاری بانجام می رسد. این رشته اضافه شده به کلید اصلی Initialization Vector یا به اختصار IV خوانده می شود. پس از این بارها این عبارت را طول انجام حملات خواهید شنید. به همین دلیل است که WEP استاندارد 64bit WEP نیز خوانده می شود (40+24=64bit). پس از گذشت مدتی از معرفی WEP استاندارد، کوتاه بودن طول کلید رمزنگاری و بالا بودن شانس شکسته شدن آن در موارد استفاده حساس، WEP ارتقاع یافته و رمزنگاری با یک کلید 104 بیتی که مجدداً یک IV با طول 24 بیت به آن اضافه می شود تحت عنوان 128bit WEP مورد استفاده قرار گرفت. برای استفاده از 64bit WEP کاربر در صورت تمایل به استفاده از کلید رمز بصورت معمولی (ASCII) می بایست عبارتی با طول 5 کاراکتر، و در صورت استفاده از 128bit WEP عبارتی با طول 13 کاراکتر (26 بیت هگز) را انتخاب کند. 256bit WEP نیز توسط برخی از تولید کنندگان پشتیبانی شده و پیاده سازی شده است. اگر چه افزایش طول کلید شانس موفقیت حمله برای شکستن کلید را کمتر می کند، اما طول محدود کلید تنها مشکل WEP نبوده است.

بمنظور اعتبارسنجی برای کنترل دسترسی و استفاده از اکسس پوینت، دو روش اعتبارسنجی در کنار WEP قابل استفاده است که Open System Authentication و Shared Key Authentication هستند. در روش OPEN، کاربر عملاً برای associate شدن به اکسس پوینت نیازی به آگاهی از کلید صحیح WEP نداشته و هر کلاینتی می تواند به اکسس پوینت متصل شود. در مرحله بعد از association یعنی شروع رمزنگاری و برقراری ارتباط در سطح شبکه است که آگاهی از کلید صحیح WEP الزامی می باشد. پس در این روش عملاً مفهوم اعتبارسنجی معنی واقعی خود را ندارد.

در روش .Shared Key Auth از کلید WEP برای اعتبارسنجی کلاینت و صدور اجازه اتصال به شبکه استفاده می شود. این اعتبارسنجی در 4 مرحله انجام می شود که به ترتیب زیر است :

1- کلاینت درخواست اعتبارسنجی را به اکسس پوینت ارسال می کند
2- اکسس پوینت یک عبارت (challenge) را بصورت شفاف (clear-text) برای کلاینت ارسال می کند
3- کلاینت می بایست این عبارت را با کلید WEP رمز کرده و به اکسس پوینت بازگرداند
4- اکسس پوینت عبارت رمز شده دریافتی از کاربر را با کلید خود رمزگشایی کرده و در صورت تطبیق آن با عبارتی که در مرحله 2 ارسال کرده، نتیجه اعتبار سنجی موفقیت آمیز بوده است.

پس از اعتبار سنجی، ادامه ارتباط و ارسال ترافیک بین کلاینت و اکسس پوینت بصورت رمز شده ادامه می یابد. در نگاه اول ممکن است از نظر امنیتی روش Shared Key امن تر بنظر برسد زیرا در روش Open عملاً اعتبارسنجی قبل از شروع رمزنگاری استفاده نمی شود، اما قضیه کاملاً برعکس است. بدلیل اینکه در روش Shared Key شانس شنود مراحل اعتبارسنجی و آگاهی از یک عبارت و مقدار رمزشده آن با کلید رمزنگاری وجود دارد، حمله برای کشف کلید رمزنگاری استفاده شده در همین ابتدا امکان پذیر می شود. زیرا از سویی شما مقدار challenge را می دانید و از سوی دیگر کلاینت، مقدار رمز شده Challenge با کلید رمزنگاری را برای اکسس

پوینت ارسال می کند که باز هم شما قادر به شنود آن خواهید بود. با دانستن عبارت clear-text و مقدار رمز شده آن دستیابی به کلید مورد استفاده برای رمزنگاری امکانپذیر خواهد شد.

مشکل WEP : اما مشکل امنیتی WEP واقعاً چیست؟! همانطور که گفته شد WEP از RC4 برای رمزنگاری استفاده می کند که یک stream cipher است. در خصوص این نوع از الگوریتم های رمزنگاری، امنیت الگوریتم منوط به **عدم** استفاده از نقطه شروع یکسان برای آغاز پروسه رمزنگاری است. به زبان بسیار ساده تر، بخشی از نقطه شروع رمزنگاری در RC4 همان کلید رمزنگاری WEP و رشته تصادفی 24 بیتی (IV) هستند که پیش از این به آنها اشاره شد. اشکال کار اینجاست که در حجم ترافیک بالا و مدت زمان طولانی، 24 بیت طول استاندارد IV برای تولید اعدادی که به اندازی کافی تصادفی باشند کافی نیست. با تولید تعداد بسیار زیادی پکت رمزنگاری شده و در نتیجه تولید یک IV برای هر پکت، در نهایت کلیه مقادیر قابل استفاده برای IV مصرف شده و پس از آن یک IV مجدداً تکرار شده و از آن برای رمزنگاری استفاده می شود. به چنین IV های تولید شده ایی اصطلاحاً Weak IV گفته می شود. این به معنی یک تهدید جدی برای RC4 و امنیت آن است ! این دقیقاً همان چیزی است که محققان برای حمله به WEP (و RC4) به دنبال آن می گشتند. مدتی پس از انتشار مقاله مربوط به ضعف های امنیتی RC4 که به آن اشاره شد، تیم دیگری از محققان روش هایی را برای منطبق کردن این ضعف ها با WEP و بررسی امکان سو استفاده از مشکلات RC4 با استفاده از ساختار و مکانیزم کاری پروتکل 802.11 و WEP در طی مقاله ایی[۳۸] منتشر کردند. پس از این تکلیف حملات و هدف اولیه آنها مشخص تر بنظر می رسد؛ پیدا کردن روش هایی برای بدست آوردن IV های تکراری بیشتر و بیشتر تا زمانی که تعداد این IV های تکراری برای انجام statistical analyze لازم برای حمله به RC4 و بدست آوردن کلید رمز کافی باشد. اما یک مشکل بزرگ بر سر راه این کار و جمع آوری تعداد Weak IV لازم وجود داشت. چرخه تکرار (collision) IV ها و تولید شدن بسته های رمزنگاری شده آنقدر کند است که در روش ها و شرایط سنتی، بسته به ترافیک شبکه گاهی ممکن بود این انتظار برای جمع شدن تعداد Weak IV لازم به چند روز و آنالیز میلیون ها پکت برسد! در حمله های ابتدایی علیه WEP و بوسیله کد ها و ابزارهای اولیه حمله، برای شکستن یک کلید 64 بیتی WEP جمع آوری 4 تا 10 میلیون پکت و آنالیز آنها برای استخراج تعداد Weak IV مورد نیاز الزامی بود. حتی در یک شبکه بسیار پر ترافیک بیسیم این کار به ساعت ها زمان نیاز دارد. بخش نهایی حمله که مربوط به RC4 و شکستن آن است پس از جمع آوری تعداد Weak IV های کافی تنها به چند دقیقه زمان نیاز دارد.

اولین پیاده سازی حملات تئوری مطرح شده علیه WEP در سال 2001 و در قالب ابزاری بنام WEPcrack[۳۹] بصورت عمومی منتشر شد. WEPcrack اولین ابزار پیاده سازی حمله به WEP نبود بلکه تحقیقی که توسط متخصص امنیتی بنام Adam Stubblefield[۴۰] در قالب مقاله ایی منتشر شد اولین پیاده سازی Practical حمله بود، اما کد ابزار وی بصورت عمومی منتشر نشد. چند روز پس از انتشار WEPcrack، گروه امنیتی Shmoo ابزار AirSnort[۴۱] را بصورت عمومی منتشر کرد که در مقایسه با WEPcrack پیاده سازی بهتر و بهینه تری از حمله WEP را در اختیار قرار گذاشت و به همین دلیل بسیاری AirSnort را بعنوان اولین ابزار این حمله شناختند. این ابزارها همگی از روش حمله ایی که بنام FMS Attack شناخته شده استفاده می کردند. AirSnort برای انجام حمله خود به جمع آوری 4 تا 6 میلیون پکت داده از شبکه بیسیم هدف احتیاج داشت. David Hulton (H1kari) در سال 2002 با انتشار یک تکنیک حمله عملی جدید تر در قالب یک مقاله[۴۲] وابزاری که مبتنی بر آن تکنیک بود (dwepcrackکه یکی از ابزارهای بسته bsd-airtools است) میزان پکت های مورد نیاز برای حمله را کاهش داد بطوری که با این روش جدید تنها به 500 هزار تا 2 میلیون پکت برای انجام حمله نیاز بود.

حملات Brute-force/Dictionary علیه WEP : روش ها و حملات ذکر شده تا این مرحله همگی جزو خانواده حملات Statistical Analysis یا اصطلاحاً Statistical Attacks قرار می گیرند. دیدگاه دیگری نیز برای حمله به WEP وجود دارد که شاید اولین راه حل نیز

بنظر برسد؛ انجام حملات Dicrionary Attack و یا Brute-force برای کشف کلید WEP. از سویی شاهد آن هستیم که در موارد زیادی کلمه ایی که بعنوان کلید WEP انتخاب شده از نظر کیفیت بسیار بد بوده و به نوعی کلمه ایی قابل حدس می باشد. بنا بر این شانس اینکه کلید WEP توسط یک فایل دیکشنری معمولی بدست بیاید زیاد است. از سویی در صورتی که بوسیله Dictionary Attack به نتیجه نرسیم، با توجه به طول 64 بیتی و یا 128 بیتی کلید WEP، زمان پردازش و محاسبه برای کنترل همه حالت های ممکن بسیار زیاد می باشد و در مورد کلید های با طول بیش از 64bit عملاً غیر قابل استفاده. با این حال این روش حمله به WEP گاهی ممکن است تنها راه ممکن باشد. شرایطی را فرض کنید که به هر دلیل ممکن شما امکان جمع آوری تعداد پکت های لازم برای انجام Statistical Attack را ندارید و یا بمنظور حفظ اختفا در طول حمله، قصد بدست آوردن کلید WEP بصورت کاملاً غیر فعالانه (Passive) را دارید. در حمله به WEP به این روش در اختیار داشتن حتی یک پکت رمزنگاری شده برای شروع کار کافی است و در طول حمله (بر خلاف روش های Statistical) نیازی به ارسال پکت از سوی حمله کننده نیست. دو ابزار WepLab[۴۳] و WepAttack[۴۴] از جمله ابزارهایی هستند که از این روش حمله پشتیبانی کرده و به شما امکان انجام آنرا می دهند. ابزار Aircrack-ng نیز اگرچه بیشتر بعنوان ابزاری برای Statistical Attacks به WEP شناخته شده اما قابلیت انجام Dictionary Attack را نیز دارد. برای استفاده از این روش حمله کافیست با روش ها و ابزارهای معرفی شده در بخش "شنود شبکه های بیسیم" همین مقاله چند پکت رمزشده توسط WEP را بدست آورده و در اختیار نرم افزار قرار دهید. در صورتی که شما خوش شانس باشید، با یک حمله Dictionary Attack ممکن است در مدت زمان کوتاهی به نتیجه برسید و دیگر نگران درگیر شدن با ابزارهای Statistical Attack نباشید! با توجه به نیاز به قدرت پردازشی بسیار بالا در این روش حمله، برخی ایده استفاده از روش های پردازشی سریعتر را در ابزارهای خود پیاده کرده اند. در مواردی که قدرت پردازشی پردازنده یک کامپیوتر به تنهایی جوابگوی نیاز محاسباتی نباشد، راه حل استفاده از Distributed Computing (استفاده از پردازنده هایی چند کامپیوتر بصورت موازی) و یا بکار گیری تکنولوژی FPGA به ذهن می رسد. در خصوص WEP نیز همین رویکرد در تعدادی از ابزارهای حمله وجود دارد. JC-WEPcrack[۴۵] یک نمونه از ابزارهایی است که به روش Distributed اقدام به حمله به WEP می کند.[46] JC-AirCrack ابزار دیگری از همان برنامه نویس است که تکنیک های حمله موجود در ابزار Aircrack (نسخه قدیمی ابزار، و نه نسخه Aircrack-NG) از جمله حمله KoreK را برای استفاده بر روی FPGA پیاده سازی کرده است. استفاده از FPGA در چنین حملاتی سرعت را بطور چشمگیری (در این مورد حداقل 20 برابر) افزایش می دهد. Pico-WEPCrack که در کنار JC-WEPcrack عرضه شده، بجای استفاده از روش Distributed از قدرت پردازشی FPGA برای کار خود استفاده کرده است.

همانطور که قابل حدس است، روش های سنتی حمله به WEP بسیار زمانبر و مشکل بوده و دردسرهای خاص خود را دارند. با توجه به قدیمی بودن این تکنیک ها وابزارهای مربوطه در این مقاله از ذکر جزئیات عملی انجام این حملات صرف نظر شده و تنها بمنظور ایجاد پیش زمینه و آگاهی لازم به آنها اشاره شد. در حال حاضر روش ها و تکنیک های مختلفی برای بهبود روال حمله, کوتاه تر و بهینه تر کردن زمان و هزینه (محاسباتی) لازم برای انجام روش های ذکر شده کشف و پیاده سازی شده که در بخش بعدی مقاله مورد بررسی قرار خواهند گرفت.

حملات پیشرفته به WEP :

حال که از مفهوم و کلیات مشکلات امنیتی RC4 و WEP آگاه شدید می توان با پیش زمینه ذهنی بهتری ابزارها و روش های حمله امروزی و مدرن برای حمله به WEP را مورد بررسی قرار داده و از میزان بهبود کیفیت حملات نسبت به روش های قدیمی آگاه شد. در بخش قبلی دیدیم که در بهترین حالت شنود و جمع آوری حداقل 500 هزار پکت الزامی بود. فراموش نکنید که بسیاری از این اعداد که در مقاله یا متن

معرفی ابزار عنوان شده اعدادی هستند که در شرایط آزمایشگاهی و نه واقعی بدست آمده اند. در حمله به یک شبکه و هدف واقعی حتی جمع آوری این تعداد پکت نیز ممکن است به چندین ساعت زمان احتیاج داشته باشد. در سال 2004 محققی با نام مستعار KoreK روش حمله جدیدی را برای حمله به WEP ابداع و آنرا منتشر کرد. ابزاری که وی برای پیاده سازی حمله خود منتشر کرد Chop-chop[۴۷] نام داشت و به همین دلیل این روش حمله در بسیاری از منابع به اسم Korek-Attack و یا Chop-chop Attack نیز شناخته می شود. این تکنیک حمله KoreK تعداد پکت های لازم برای شکستن WEP را به 300,000 کاهش داد. اگر چه این عدد در مقایسه با حداقل 4 میلیون پکت در روش های اولیه حمله بسیار چشمگیر است، اما این باعث نشد محققین روش های بهتری را ابداع نکنند. اساس و روش کار حمله Korek کمی پیچیده است. اگر بخاطر داشته باشید گفتیم که WEP برای حفظ محرمانگی داده ها از RC4 و برای کنترل صحت داده های دریافتی از الگوریتم CRC-32 استفاده می کند. نقطه ضعف مورد استفاده KoreK برای حمله نیز پیاده سازی نا امن CRC-32 در WEP است که بنام ICV[۴۸] شناخته می شود.

KoreK (Chop-Chop) Attack : روال حمله Korek را به زبان بسیار ساده اینطور میتوان شرح داد که؛ ابتدا یک پکت رمز شده شنود می شود. سپس یک بایت از پکت تغییر داده شده و با توجه به اینکه Integrity هر پکت بوسیله CRC-32 کنترل می شود، مقدار CRC hash قبلی (اصلی) پکت بدلیل تغییر ما غیر معتبر خواهد بود. بنابراین می بایست این مقدار مجدداً محاسبه شده و در پکت درج شود. در شرایط عادی این کار بسیار ساده است، اما با توجه به اینکه ما متن و محتوایی که قرار است از آن CRC Hash تولید کنیم را نمی دانیم، این کار مشکل و پیچیده می شود و نکته حمله Korek نیز در همین قسمت کار است، یعنی روشی برای تغییر محتوای بسته های رمز شده با WEP در عین حفظ Integrity آنها و البته بدون نیاز به آگاهی از کلید رمز نگاری و یا محتوای رمز شده. با توجه به اینکه ICV درج شده در بسته در واقع خروجی یک الگوریتم XOR است بنا بر این با روش های آزمایش و خطا شانس حدس زدن مقدار صحیح از طریق Brute-force وجود دارد. گفته شد که ما تنها یک بایت از پکت را تغییر داده و ICV جدید را بر اساس آن محاسبه ودر پکت درج می کنیم. شانس اینکه حدس ما و ICV محاسبه شده درست باشد 1 در 255 (حالت ها و مقدار های ممکن برای یک بایت تغییر داده شده) است. پس از تولید پکت ویرایش شده با ICV جدید، این پکت به اکسس پوینت ارسال می گردد. بر طبق استاندارد WEP و کنترل CRC-32 اگر مقدار محاسبه شده ما اشتباه بوده باشد پکت نادیده گرفته شده و Drop می شود اما اگر حدس ما در محاسبه درست بوده باشد، اکسس پوینت آن پکت را پس از دریافت از ما، به شبکه ارسال خواهد کرد. با توجه به اینکه حمله کننده در حال شنود ترافیک می باشد، ارسال این پکت خاص توسط اکسس پوینت را شناسایی کرده و تشخیص می دهد. بدین ترتیب بدون آگاهی از مقدار واقعی کلید رمزنگاری WEP ما موفق به حدس یک بایت از محتوای پکت رمز شده گشتیم. همین روال و چرخه ذکر شده برای تک تک بایت های پکت مد نظر ما تکرار می شود با ذکر این نکته که هر بار پس از دریافت پکت معتبر تشخیص داده شده از نظر اکسس پوینت، پکت ارسالی از آن برای حدس بایت بعدی و تکرار چرخه مورد استفاده قرار می گیرد. در نهایت ما کل آن پکت را بدون در اختیار داشتن کلید رمزنگاری، رمزگشایی کرده و می توانیم از محتویات آن آگاه شویم. در واقع خود اکسس پوینت بایت به بایت پکت را برای ما رمزگشایی کرده است. برای آگاهی از جزئیات فنی و کامل این حمله میتوانید به مستند ارائه شده در بسته نرم افزار Chop-Chop و یا مقاله ایی[۴۹] که به تشریح روش کار آن پرداخته مراجعه کنید. با توجه به استفاده از تئوری این حمله در بسیاری از حملات جدید، توصیه می شود حتماً جزئیات کامل این روش حمله مطالعه گردد.

در نگاه اول حمله KoreK ممکن است چندان هم کاربردی بنظر نرسد، با توجه به اینکه در هر حمله شما تنها قادر به رمزگشایی محتوای یک پکت هستید. همچنین حدس بایت به بایت محتوای یک پکت بسیار زمانبر و کند است مخصوصاً اگر حجم محتوای پکت زیاد باشد. پس حمله KoreK چطور تا این حد می تواند به ما کمک کند؟ جواب این است که عموماً از خود حمله KoreK به تنهایی برای حمله به یک شبکه بیسیم محافظت شده توسط WEP استفاده نمی شود. بلکه از آن برای بهبود سرعت انجام نوع دیگری از حملات استفاده می شود که در ادامه مورد بررسی قرار می گیرند. حمله Korek در برخی شرایط خاص تنها راه شروع حمله برای استخراج کلید رمزنگاری WEP می باشد و در

صورت عدم استفاده از آن، حمله اصلی ممکن است بسیار زمانبر و در اغلب موارد حتی غیر ممکن گردد. از جمله این شرایط خاص می توان به زمانی اشاره کرد که اکسس پوینت هیچ کلاینتی که به آن متصل شده باشد نداشته و ترافیک زیادی نیز تولید نمی کند که بتوان بصورت غیر فعالانه اقدام به جمع آوری Weak IV ها در زمان معقول کرد.

گفتیم که حمله Korek بایت به بایت محتویات پکت را رمزگشایی می کند. بنابر این حمله در خصوص پروتکل هایی که حجم پکت های آنها کوچک است کارایی داشته و برخی از پروتکل ها نیز دارای محتوای تقریباً مشخصی بوده و در پکت های ارسالی قسمت زیادی از محتوا همواره ثابت و قابل پیشبینی است. یکی از مناسب ترین گزینه ها برای حمله، پروتکل ARP و پکت های تولید شده آن هستند. زیرا اولاً دارای حجم کم و محتوای قابل پیشبینی هستند و ثانیاً ماهیت این پروتکل و نحوه کار آن به ما امکان پیاده سازی حملات Spoofing را نیز در سطح شبکه می دهد. تا این قسمت از مطلب را در ذهن خود حفظ کنید. از سوی دیگر مطرح شد که مشکل اصلی حمله به WEP در روش های سنتی این است که حمله کننده می بایست آنقدر منتظر بماند تا Weak IV ها و تعداد پکت های داده لازم جمع آوری شوند و سپس اقدام به حمله Statistical به RC4 شود. در یک شبکه شلوغ و پر ترافیک نیز این کار به ساعت ها زمان نیاز دارد، چه رسد به یک شبکه معمولی و یا کم ترافیک! یکی از راه حل هایی که برای این مشکل بنظر رسیده و مورد استفاده قرار میگیرد، وادار کردن سیستم های متصل به شبکه (اکسس پوینت و کلاینت های متصل به آن) به تولید ترافیک بیشتر است. یکی از موثر ترین روش ها برای انجام این کار، پیاده سازی حملات ARP Spoofing است بطوری که یک درخواست ARP جعل شده از جانب یکی از کلاینت های شبکه به اکسس پوینت ارسال شده و آنرا وادار به پاسخ دادن به آن کند. اما با توجه به وجود WEP چطور می توان چنین بسته ایی را بصورت معتبر تولید کرد؟ اگر ما از کلید رمزنگاری برای تولید یک بسته معتبر اطلاع داشتیم، دیگر نیازی به حمله به شبکه نبود! حالا وقت آن است که مجدداً به تئوری مطرح شده در مورد Korek بازگردید. بکمک حمله Korek میتوان براحتی یک پکت WEP را شنود کرده، محتوای آنرا رمزگشایی کرده و پس از آن پکت جدیدی را با محتوای دلخواه تولید و ارسال کرد، بدون اینکه نیازی به آگاهی از کلید رمزنگاری WEP باشد. این دقیقاً همان چیزی است که به آن نیاز داشتیم! شاید تصور کنید که بدون حمله KoreK نیز تشخیص بسته های ARP (با توجه به اینکه Header پکت های ارسالی در شبکه های بیسیم رمزنگاری نمی شوند) و ارسال مجدد آنها به اکسس پوینت امکانپذیر است. درست. اما اولاً شما قادر به تغییر محتوای آنها نیستید و ثانیاً اگر کلاینتی در شبکه وجود نداشته باشد که شما درخواست ARP او را جعل و مجدداً ارسال کنید (سناریو اکسس پوینت بدون کلاینت) ممکن است انتظار شما برای جمع آوری Weak IV بیش از حد طولانی شود. از سوی دیگر در صورت وجود مکانیزم امنیتی Dynamic WEP که کلید WEP را پس از گذشت مدت زمان و شرایط خاص بصورت Dynamic تغییر می دهد، طولانی شدن زمان انتظار مساوی است با تغییر کلید WEP ایی که شما در حال جمع آوری IV برای شکستن آن بوده اید. این حمله روشی است که به شما امکان مقابله با Dynamic WEP را نیز می دهد زیرا زمان حمله آنقدر کوتاه است که شانس شما برای رمزگشایی یک پکت قبل از تغییر کلید WEP بصورت خودکار بسیار بالاست و تقریباً همیشه به نتیجه می رسد.

برای انجام حمله Korek می بایست ابتدا کارت در حالت Monitor قرار گرفته و اطلاعات اولیه در خصوص اکسس پوینت مورد نظر بدست آید. برای این کار می توان از ابزار Airmon-ng برای تغییر حالت کارت و سپس از ابزار Airodump-ng برای شنود و استخراج اطلاعات و در نهایت از ابزار Aireplay-ng برای حمله استفاده کرد. Aireplay-ng علاوه بر اینکه تمام قابلیت های نرم افزار اصلی chopchop را در بر دارد، برخی از قابلیت های آن و همچنین روش حمله را نیز بهبود بخشیده و مجدداً پیاده سازی کرده است. در صورتی که کلاینت دیگری بجز حمله کننده در حال استفاده از اکسس پوینت باشد می توان از ترافیک تولید شده توسط آن و مک آدرس سیستم مورد نظر استفاده کرد، در غیر اینصورت می بایست قبل از حمله KoreK با استفاده از حمله Fake Authentication[50] نفوذگر سیستم خود را به اکسس پوینت متصل و معرفی کند تا اکسس پوینت به وی اجازه ارسال پکت بدهد. انجام حمله Fake Authentication پیش نیاز انجام بسیاری از

حملات به اکسس پوینت می باشد. برای Assosiate شدن به یک اکسس پوینت و ارسال پکت به آن نیازی به دانستن اطلاعاتی در مورد شبکه هدف یا حتی کلید رمزنگاری نیست. کافیست با استفاده از قابلیت مربوطه، ابزار Aireplay-ng بصورت زیر مورد استفاده قرار گیرد :

```
aireplay-ng -1 5 -e name -a 00:14:6C:7E:40:80 -h 00:09:5B:EC:EE:F2 ath0
```

سویچ "1-" معرف حمله Fake Authentication و "5" معرف تکرار حمله هر 5 ثانیه یکبار است. "e-" معرف نام شبکه بیسیم و "a-" معرف مک آدرس اکسس پوینت (bssid) می باشد. "h-" نیز معرف مک آدرس سیستم شما و یا یک مک آدرس جعلی که شما برای خود انتخاب کرده و توسط ابزارهایی مانند macchanger آنرا مورد استفاده قرار داده اید. در صورت موفقیت آمیز بودن حمله شما خروجی مشابه زیر دریافت خواهید کرد.

```
18:18:20  Sending Authentication Request
18:18:20  Authentication successful
18:18:20  Sending Association Request
18:18:20  Association successful :-)
```

MAC Address Spoofing *: توصیه می شود همواره برای حمله از مک آدرس های جعلی استفاده کنید. همچنین در صورتی که این حمله به نتیجه نرسد، ممکن است علت آن استفاده از مکانیزم امنیتی MAC Filtering در اکسس پوینت باشد، که در چنین شرایطی اکسس پوینت تنها به سیستم های دارای مک آدرس از پیش معرفی شده اجازه اتصال می دهد. برای فرار از این محدودیت بهترین و ساده ترین راه شنود ترافیک اکسس پوینت مد نظر و بدست آوردن چند مک آدرس از سیستم هایی است که با موفقیت به اکسس پوینت متصل شده و پکت های اطلاعاتی (و نه فقط پکت های Beacon) ارسال کرده اند. در حمله MAC Add. Spoofing شما با تغییر مک آدرس کارت شبکه خود به یکی از آدرس های مجاز و تعریف شده در اکسس پوینت براحتی قادر به دور زدن این محدودیت اعمال شده خواهید بود. در سیستم عامل ویندوز این کار مستلزم استفاده از ابزارها و برنامه های جانبی تهیه شده به همین منظور است، اما در سیستم عامل لینوکس در صورت عدم تمایل به استفاده از ابزارها و اسکریپت های آماده حتی بوسیله دستورات خود سیستم عامل نیز می توان این کار را انجام داد. تنها نکته این است که تغییر مک آدرس می بایست قبل از اتصال (Associate شدن) به اکسس پوینت صورت گیرد.*

پس از انجام موفقیت آمیز Fake Authentication می توان با دستور زیر حمله KoreK را آغاز کرد:

```
aireplay-ng -4 -h 00:09:5B:EC:EE:F2 -b 00:14:6C:7E:40:80 ath0
```

در این مثال "4-" ابزار را برای استفاده از حمله KoreK تنظیم میکند، "h-" معرف مک آدرس سیستم نفوذگر و یا یک کلاینت دیگر متصل به اکسس پوینت است. "b-" مشخص کننده BSSID یا همان مک آدرس اکسس پوینت و در نهایت "ath0" نام اینترفیس مورد استفاده است. عموماً پس از مدت زمانی کوتاه (و شاید طولانی، در یک شبکه بدون ترافیک!) ابزار به محض موفقیت در شنود یک پکت رمز شده با WEPبه شما پیغام داده و پیش از استفاده از آن پکت برای حمله از شما تآیید می گیرد. ممکن است سوال کنید که آیا زمانی هم هست که می بایست پکتی را تآیید نکرد؟ بله. حمله مد نظر ما در مورد پکت هایی با حجم کم سرعت وکارایی بالایی دارد، پس بهتر است پکت انتخابی ما سایز کم و قابل قبولی داشته باشد (بین 30 تا 80 بایت داده). در صورت شنود یک پکت ARP انتخاب ما نیز بی تردید همان بسته است. پکت های ARP را از روی آدرس مقصد آنها که FF:FF:FF:FF:FF:FF است می توان شناسایی کرد. پس از آن Aireplay-ng حمله را آغاز کرده و در نهایت دو فایل خروجی شامل پکت رمزگشایی شده و فایل دیگری با پسوند.xor حاوی PRGA مورد استفاده در رمزنگاری تولید می کند که ما در ادامه از آن برای تولید پکت های جدید استفاده خواهیم کرد. خروجی ابزار در این روال بصورت زیر است :

```
Read 165 packets...

    Size: 86, FromDS: 1, ToDS: 0 (WEP)

    BSSID  =  00:14:6C:7E:40:80
    Dest. MAC  =  FF:FF:FF:FF:FF:FF
    Source MAC  =  00:40:F4:77:E5:C9

    0x0000:  0842 0000 ffff ffff ffff 0014 6c7e 4080 .B..........l~@.
    0x0010:  0040 f477 e5c9 603a d600 0000 5fed a222 .@.w..`:...._.."
    0x0020:  e2ee aa48 8312 f59d c8c0 af5f 3dd8 a543 ...H....._=..C
    0x0030:  d1ca 0c9b 6aeb fad6 f394 2591 5bf4 2873 ....j.....%.[.(s
    0x0040:  16d4 43fb aebb 3ea1 7101 729e 65ca 6905 ..C...>.q.r.e.i.
    0x0050:  cfeb 4a72 be46                          ..Jr.F

Use this packet ? y

Saving chosen packet in replay_src-0201-191639.cap

Offset   85 ( 0% done) | xor = D3 | pt = 95 |  253 frames written in   760ms
Offset   84 ( 1% done) | xor = EB | pt = 55 |  166 frames written in   498ms
Offset   83 ( 3% done) | xor = 47 | pt = 35 |  215 frames written in   645ms
.
.
.
Offset   37 (92% done) | xor = 13 | pt = 01 |  232 frames written in   695ms
Offset   36 (94% done) | xor = 83 | pt = 00 |   19 frames written in    58ms
Offset   35 (96% done) | xor = 4E | pt = 06 |  230 frames written in   689ms
Sent 957 packets, current guess: B9...

The AP appears to drop packets shorter than 35 bytes.
Enabling standard workaround: ARP header re-creation.

Saving plaintext in replay_dec-0201-191706.cap
Saving keystream in replay_dec-0201-191706.xor

Completed in 21s (2.29 bytes/s)
```

مرحله بعد در حمله، تولید یک پکت جدید با محتوای دلخواه برای ارسال به اکسس پوینت است، بدون اینکه ما از کلید WEP آگاهی داشته باشیم. خروجی مرحله قبل (فایل.xor) حاوی اطلاعات لازم برای انجام این کار است. پکت تولید شده ما یک درخواست ARP خواهد بود که به اکسس پوینت دائماً ارسال خواهد شد تا وی را مجبور به ارسال پاسخ و در نتیجه تولید بیشتر و بیشتر پکت (حتی در یک شبکه کاملاً خلوت) کند، و این کار سبب خواهد شد تا ما در مدت زمان کوتاهی قادر به جمع آوری تعداد پکت ها و Weak IV های لازم با شیم. برای ساختن پکت جعلی جدید از ابزار packetforge-ng بصورت زیر استفاده می شود :

```
packetforge-ng -0 -a 00:14:6C:7E:40:80 -h 00:09:5B:EB:C5:2B -k 255.255.255.255 -l 255.255.255.255 -y replay_dec-0627-022301.xor -w arp.cap
```

بدین ترتیب ما یک پکت جدید ARP تولید خواهیم کرد که اکسس پوینت پس از دریافت، آنرا بصورت broadcast مجدداً ارسال خواهد کرد زیرا آدرس مقصد "-k" پکت 255.255.255.255 تنظیم شده است. "-a" همانند قبل معرف مک آدرس اکسس پوینت و "-h" معرف مک آدرس سیستم ما یا سیستمی که در مراحل قبل از مک آدرس آن استفاده کرده ایم است. فراموش نکنید که مک آدرس مورد استفاده بعنوان کلاینت می بایست مربوط به سیستمی باشد که به اکسس پوینت متصل و Associate شده است. بمنظور ایجاد این پکت توسط سویچ "-y" فایل.xor تولید شده در مرحله قبل می بایست به ابزار معرفی شود. در نهایت پکت تولید شده در فایل arp.cap ذخیره خواهد شد.

ما این پکت را به اکسس پوینت بارها ارسال کرده و به موازات آن توسط ابزار Airodump-ng اقدام به شنود پکت های تولید شده و ذخیره آنها برای حمله statistical نهایی می کنیم. ابتدا Airodump را توسط دستوری مانند مثال زیر اجرامی کنیم :

```
Airodump-ng –w output –channel 6 –bssid {AP MAC} ath0
```

کانالی که Airodump-ng بر روی آن شنود می کند می بایست کانالی باشد که اکسس پوینت مورد نظر بر روی آن فعال است، همچنین بر خلاف حالت شنود پیش فرض که Airodump بین کانال های مختلف پرش می کند، بمنظور جلوگیری از از دست دادن پکت های دریافتی، کانال شنود توسط سوئیچ "channel—" ثابت می شود. سوئیچ "bssid—" پکت های دریافتی را فیلتر کرده و تنها پکت های ارسالی از اکسس پوینت مورد نظر ما را ذخیره می کند. در آخرین مرحله زمان ارسال پکت ساخته شده فرا می رسد که توسط ابزار Aireplay-ng بصورت زیر انجام می شود. "2-" برای ارسال پکت دلخواه بصورت Interactive استفاده شده و توسط "r-" فایل حاوی پکت که قبلاً توسط ابزار packetforge-ng ساخته شد به ابزار تغذیه می گردد.

```
aireplay-ng -2 -r arp.cap ath0
```

پس از اجرای این دستور به موازات airodump-ng خواهیم دید که تعداد پکت های داده ارسالی از اکسس پوینت به سرعت شروع به افزایش می کنند بطوری که در عرض چند دقیقه شما قادر به جمع آوری تعداد پکت های لازم برای استخراج کلید WEP خواهید بود. جمع آوری حدوداً 20000 پکت داده (که در واقع بیش از 90٪ این پکت ها حاوی IV های مد نظر ما هستند) برای شکستن یک کلید 64bit کفایت می کند. با جمع شدن این تعداد پکت شما می توانید بدون نیاز به متوقف کردن airodump، ابزار Aircrack-ng را بر روی فایل خروجی تولید شده توسط airodump اجرا کنید و در صورتی که با گذشت چند دقیقه ابزار موفق به شکستن کلید نشد، این کار را مجدداً پس از جمع آوری شدن چند هزار پکت دیگر تکرار کنید. در شرایط ایده آل و در دسترس بودن تعداد کافی IV، عمل شکستن کلید توسط Aircrack-ng (در آخرین نسخه ابزار) تنها چند ثانیه طول خواهد کشید. برای استفاده از آن کافیست ابزار Airodump-ng را بصورت " Aircrack-ng output.cap " اجرا کرده و منتظر نتیجه بمانید. "output.cap" همان فایل خروجی ابزار Airodump-ng می باشد که در مسیر جاری ذخیره می شود.

نکته آخر اینکه همیشه امکان استفاده از حمله Korek وجود ندارد و برخی از اکسس پوینت ها نسبت به این حمله آسیب پذیر نمی باشند. در چنین شرایطی میتوان امکان انجام حمله Fragmentation را بررسی کرد. یکی از روش های تدافعی اکسس پوینت ها برای مقابله با این حمله، بلاک کردن کلاینت پس از تعداد مشخصی تلاش برای ارسال بسته های با CRC اشتباه است. ضعف های امنیتی که KoreK بر آنها برای حمله استفاده می کند در WPA تا حد زیادی مرتفع شده است. امکان انجام این حمله بر روی WPA در بخش بعدی مقاله و حملات جدید علیه WPA مورد بررسی قرار خواهد گرفت و علت توضیح بیشتر در مورد این حمله به نسبت سایر حملات در این بخش مقاله، نیاز به ارجاع به آن در بخش های بعدی بوده است.

Fragmentation Attack : در سال 2006 تکنیک حمله جدیدی علیه WEP معرفی شد که بخشی از نتایج حاصله از حمله از نظر کاربرد و کارایی مشابه حمله KoreK می باشد اما روش انجام آن کاملاً متفاوت است. جزئیات این حمله تحت مقاله ایی بنام " The final nail in "WEP`s coffin"[51] منتشر شد که تکنیک ها و ایده های حمله جالبی در قالب آن، برای اولین بار منتشر شدند. با انجام حمله Fragmentation[52] می توان یک بسته رمزنگاری شده توسط WEP را در مدت زمان بسیار کمتر نسبت به حمله KoreK رمزگشایی کرده و 1500بایت از PRGA استفاده شده برای رمزنگاری پکت را استخراج کرد. همچنین توسط این حمله در شرایط خاص (که فراهم کردن آن چندان هم مشکل نیست) امکان رمزگشایی پکت های WEP بصورت Real-time فراهم می شود. برای استفاده از این تکنیک اکسس پوینت

می بایست به اینترنت دسترسی داشته و حمله کننده نیز یک سیستم تحت کنترل خود در اینترنت در اختیار داشته باشد که اکسس پوینت بتواند با آن ارتباط برقرارا کند. حمله Fragmentation حتی بر روی اکسس پوینت هایی که کلید WEP را بصورت Dynamic تغییر می دهند نیز کارساز بوده و راهکار امنیتی Dynamic WEP که برخی از تولید کننده گان برای پوشش دادن ضعف های امنیتی WEP معرفی کردند نیز در مقابل این حمله آسیب پذیر است. علت این امر سرعت بالای این حمله در بدست آوردن PRGA پیش از اینکه کلید WEP بصورت dynamic تغییر کند است. این حمله بر خلاف KoreK که بر روی الگوریتم رمزنگاری و مشکل پیاده سازی CRC-32 تمرکز داشت، بر روی ضعف های پیاده سازی خود پروتکل 802.11 تکیه دارد و اساس آن بر ارسال درخواست های جعلی ARP و تحلیل پاسخ های دریافتی از هدف برای استخراج PRGA است. علت استفاده از پکت های ARP مشخص بودن و قابل حدس بودن بخش زیادی از محتوای هر پکت است و اینکه تنها چند بیت از یک بسته ARP در حال تغییر می باشد، و این موضوع امکان پیاده سازی حملات Plain-text Attack به الگوریتم رمزنگاری را برای استخراج داده و PRGA مورد استفاده در رمزنگاری فراهم می کند. منشآ نام حمله نیز (سو)استفاده از مکانیزم Fragmentation پیاده سازی شده در پروتکل 802.11 است. در صورتی که با مفهوم Fragmentation آشنا نیستید بطور خلاصه میتوان آنرا روش و استانداردی برای شکستن (قطعه قطعه کردن) یک پکت و ارسال محتوای آن (Packet Payload) به مقصد در قالب چند پکت مجزا عنوان کرد، بطوری که گیرنده این چند پکت مجزا را پس از دریافت مجدداً با هم ترکیب کرده و با کل آنها به دید یک پکت واحد برخورد خواهد کرد. ضعف در پیاده سازی Fragmentation در کنار استفاده از WEP در پروتکل 802.11 سبب می شود که حمله کننده بتواند با fragment کردن یک بسته رمز شده به قطعات 8 بایتی و ارسال حداکثر 16 بسته بدین صورت با یک IV واحد، موفق به تزریق 64 بایت داده شده و پس از تحلیل پاسخ اکسس پوینت به این پکت ها حداقل 8 بایت از PRGA مورد استفاده بدست خواهد آمد. ضعف پیاده سازی fragmentation در 802.11 نیز در همین نکته است که با هر یک از این fragment های ارسالی بطور مجزا برخورد کرده و در پاسخ های ارسالی به ازای هر یک از آنها یک IV جدید تولید کرده و پاسخ را رمزنگاری می کند. برای شروع و انجام حمله Fragmentation دریافت حداقل و تنها یک پکت حاوی اطلاعات (و نه Beacon) از اکسس پوینت الزامی است. حتی در شرایط معمولی و حتی شبکه های بسیار کم ترافیک مانند شبکه های خانگی نیز زمان انتظار برای دریافت چنین پکتی بسیار کوتاه خواهد بود. این حمله البته در مقایسه با KoreK نیازمندی های خاصی نیز دارد. بطور مثال تمامی پکت های ارسالی در طول حمله حتماً می بایست به درستی توسط اکسس پوینت دریافت شده و نتیجه آنها شنود شود. گم شدن هر بسته به معنی شکست خوردن کل حمله است، و این به این معنی است که برای انجام این حمله داشتن دید رادیویی با کیفیت نسبت به اکسس پوینت الزامی می باشد. همچنین Associate شدن به اکسس پوینت پیش نیاز الزامی این حمله است. در نهایت تعداد پکت های ارسالی برای انجام حمله Fragmentation عموماً بسیار بیشتر از حمله KoreK می باشد، اگرچه در قبال این مورد شما مزایای بسیار زیادی نیز بدست خواهید آورد که توسط حمله Korek یا حملات دیگر به آنها دست نخواهید یافت. برای آگاهی از این جزئیات خواندن مقاله اصلی مربوط به حمله توصیه می شود، زیرا مقاله مورد بحث گویا بوده و بر خلاف بسیاری از مقالات مشابه برای عموم نیز تا حد زیادی قابل استفاده است. بنا بر این نیازی به بازنویسی مقاله مذکور در قالب این متن نیست.

روش استفاده از این حمله نیز مشابه Korek است. پس از انجام مراحل اولیه از جمله Fake Authentication مشابه آنچه برای حمله KoreK توضیح داده شد، کافی است از ابزار Aireplay-ng بصورت زیر استفاده شود :

```
aireplay-ng -5 -b 00:14:6C:7E:40:80 -h 00:0F:B5:AB:CB:9D ath0
```

فکر می کنم توضیح زیادی لازم نباشد. "5-" معرف حمله Fragmentation و "b-" معرف مک آدرس اکسس پوینت می باشد. خروجی این حمله عیناً مشابه KoreK بوده و فایلxor. حاوی PRGA استخراج شده در صورت موفقیت آمیز بودن حمله تولید خواهد شد. پس از تولید این فایل، روال استفاده از آنها برای تولید پکت جدید مشابه موارد توضیح داده شده برای حمله KoreK می باشد. کافیست پکت جدید را

توسط packetforge-ng تولید کرده و با استفاده از aireplay-ng -2 آنرا ارسال کنیم و به موازات آن توسط Airodump-ng اقدام به شنود و ذخیره سیل پکت های تولید شده کنیم. خروجی اجرای حمله Fragmentation مشابه مثال زیر می باشد :

```
aireplay-ng -5 -b 00:14:6C:7E:40:80 -h 00:0F:B5:AB:CB:9D ath0

Waiting for a data packet...
Read 96 packets...

    Size: 120, FromDS: 1, ToDS: 0 (WEP)

        BSSID  =  00:14:6C:7E:40:80
     Dest. MAC  =  00:0F:B5:AB:CB:9D
    Source MAC  =  00:D0:CF:03:34:8C

    0x0000:  0842 0201 000f b5ab cb9d 0014 6c7e 4080 .B.........l~@.
    0x0010:  00d0 cf03 348c e0d2 4001 0000 2b62 7a01 ....4...@...+bz.
    0x0020:  6d6d b1e0 92a8 039b ca6f cecb 5364 6e16  mm.......o..Sdn.
    0x0030:  a21d 2a70 49cf eef8 f9b9 279c 9020 30c4 ..*pI.....'.. 0.
    0x0040:  7013 f7f3 5953 1234 5727 146c eeaa a594  p...YS.4W'.l....
    0x0050:  fd55 66a2 030f 472d 2682 3957 8429 9ca5 .Uf...G-&.9W.)..
    0x0060:  517f 1544 bd82 ad77 fe9a cd99 a43c 52a1  Q  .D...w.....<R.
    0x0070:  0505 933f af2f 740e                    ...?./t.

Use this packet ? y


Saving chosen packet in replay_src-0124-161120.cap
Data packet found!
Sending fragmented packet
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 384 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 1500 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Saving keystream in fragment-0124-161129.xor
Now you can build a packet with packetforge-ng out of that 1500 bytes keystream
```

ARP Request Replay Attack : این حمله شاید به نوعی موثر ترین و سریع ترین روش ممکن برای وادار کردن اکسس پوینت به تولید پکت های داده (و در نتیجه IV) بیشتر است تا بتوان با سرعت هر چه بیشتر تعداد پکت های لازم برای Statistical Attack به الگوریتم RC4 و شکستن کلید رمز WEP را بدست آورد. در شرایط مناسب سرعت تزریق پکت ها و وادار کردن اکسس پوینت به تولید پکت می تواند به 500 پکت در ثانیه برسد (این عدد وابستگی مستقیم به قدرت و کیفیت سیگنال و همچنین کارت مورد استفاده برای Packet injection دارد)، بنا بر این شما قادر خواهید بود در کمتر از 1 دقیقه تعداد پکت ها و Weak IV های لازم برای شکستن WEP را بدست آورید! اساس کار حمله ARP Request Replay[۵۳] در عین سادگی فوق العاده موثر است. روش کار بدین صورت است که با شنود ترافیک، به محض دریافت یک بسته ARP از ترافیک شبکه آنرا مجدداً به اکسس پوینت ارسال می کنیم. این کار سبب می شود اکسس پوینت نیز بسته دریافتی و تزریق شده را دوباره منتشر کند، البته با یک IV جدید. بنا بر این با هر باز تکرار این کار یک IV جدید بدست می آید. انجام این چرخه ارسال مجدد تا زمان جمع آوری پکت ها و IV های لازم برای حمله به WEP ادامه می یابد. ممکن است این سوال برای شما مطرح شود که در این روش چطور بدون آگاهی از محتویات و جزئیات یک پکت رمز شده می توان نوع آنرا تشخیص داد؟ جواب ساده است. در پروتکل 802.11 و در زمان استفاده از WEP تنها Payload پکت است که رمز می شود و عنوان (Header) پکت که مشخص کننده ارسال کننده و دریافت کننده پکت است دست نخورده باقی می ماند. با توجه به مقصد خاص مورد استفاده در پکت های ARP (مک آدرس گیرنده پکت

در arp همواره FF:FF:FF:FF:FF:FF است) و این نکته که حجم پکت های arp عموماً ثابت و برابر 36 بایت است، تشخیص این پکت ها حتی در صورت استفاده از WEP نیز براحتی میسر است. حتی می توان این پکت ها را بدون نیاز به آگاهی از محتوای (Payload) رمز شده آنها تنها با تغییر آدرس فرستنده در header جعل کرده و مجدداً ارسال کرد.

سادگی مکانیزم این حمله سبب سهولت انجام آن نیز می شود. برای انجام این حمله کافیست حمله کننده یا خود به اکسس پوینت متصل شود (با استفاده از حمله Fake Authentication که مورد بررسی قرار گرفت) و یا با شنود ترافیک و بدست آوردن مک آدرس یک کلاینت فعال و متصل به اکسس پوینت، از آن برای حمله استفاده کند. حالت دوم دارای اولویت بوده و منجر به نتیجه بسیار بهتری خواهد شد زیرا در صورت عدم وجود ترافیک و کلاینت متصل به شبکه، و در نتیجه عدم وجود پکت های ARP ارسالی به اکسس پوینت، شما ناچار به استفاده از حمله Fragmentation و یا KoreK برای استخراج PRGA از یک پکت داده خواهید بود(که می تواند از نوع ARP هم نباشد) و در نهایت تولید یک پکت ARP با استفاده از PRGA بدست آمده.

پس از انجام Fake Authentication و مجاز شدن برای ارسال پکت به اکسس پوینت، به موازات اجرای نرم افزار Airodump-ng برای شنود ترافیک و جمع آوری IV های تولید شده، کافیست ابزار Aireplay-ng را بصورت زیر مورد استفاده قرار دهیم :

```
aireplay-ng -3 -b 00:13:10:30:24:9C -h 00:11:22:33:44:55 ath0
```

در این حالت، "3-" معرف حمله ARP Request Reply و "b-" معرف مک آدرس اکسس پوینت (BSSID) می باشد. "h-" نیز در صورتی که از Fake Authentication استفاده کرده باشید مک آدرس سیستم شما، و در صورت هدف قرار دادن کلاینت های دیگر، مک آدرس کلاینتی است که در حال حاضر به اکسس پوینت متصل است. با شروع Packet Injection توسط Aireplay-ng نتیجه کار به وضوح در گزارشات خروجی ابزار Airodump-ng قابل مشاهده خواهد بود. در نهایت و پس از جمع آوری تعداد لازم IV زمان استفاده از ابزار Aircrack-ng فرا می رسد. در شرایط واقعی و یک حمله در محیط بیرون این حمله از ابتدا تا انتها و شکسته شدن کلید WEP ممکن است بین 5 تا 10 دقیقه زمان نیاز داشته باشد، اگرچه در محیط آزمایشگاهی این زمان ممکن است از یک دقیق تجاوز نکند. همچنین در صورت استفاده از کلید های 128 یا 256 بیتی شما نیاز به جمع آوری IV های بیشتری دارید.یک کلید 128 بیتی براحتی و در عرض چند ثانیه با جمع آوری 40000 پکت داده قابل شکستن است.

روش های دیگری نیز برای شرایط متفاوت و مختلف مقابله با WEP وجود دشته و در ابزار Aireplay-ng پیاده سازی شده اند (Caffe-latte attack & Cfrag attack) که بدلیل شباهت روال انجام آنها با موارد عنوان شده، تنها به معرفی و کاربرد آنها بسنده می کنم.

Caffe-latte Attack : در شرایط عادی برای انجام حمله می بایست در محدوده پوشش رادیویی اکسس پوینت و یک کلاینت متصل به آن قرار داشت تا بتوان حملات را انجام داد. بوسیله این تکنیک حمله که در ابزار Aireplay-ng توسط سوئیچ "6-" قابل استفاده است، میتوان حمله و بدست آوردن IV های لازم برای شکستن WEP را حتی در صورتی که فقط کلاینت متصل به اکسس پوینت در محدوده ما قرار داشته باشد انجام داد. یعنی در این حمله، نیازی به قابلیت دریافت سیگنال از اکسس پوینت نیست و همین که یک کلاینت متصل به آن در دامنه دید رادیویی ما باشد کفایت می کند.

Cfrag Attack : همیشه هدف شما از شکستن WEP اتصال به اکسس پوینت نیست. در برخی موارد ممکن است هدف شکستن کلید WEP مورد استفاده بین دو کلاینت باشد که با روش Ad-HOC (اتصال مستقیم، بدون اکسس پوینت) به یکدیگر متصل شده اند. در چنین حالتی بوسیله این تکنیک حمله از پکت های IP و یا ARP ارسالی یک کلاینت پس از دستکاری، علیه خود آن استفاده می شود تا بتوان وی

را مجبور به تولید IV های بیشتر و لازم برای شکستن کلید WEP کرد. از این روش همچنین می توان برای حمله به اکسس پوینت های نرم افزاری و کلاینت های آن استفاده کرد. این حمله در نرم افزار Aireplay-ng توسط سوئیچ "7-" فعال می گردد.

PTW Attack: در سال 2007 یک بار دیگر امنیت WEP مورد چالش قرار گرفت و روش های حمله پیشین مورد بازبینی قرار گرفت تا شاید بتوان حملات بهینه تری را پیاده سازی کرد. اگر به خاطر داشته باشید عنوان شد که نقطه شروع حملات به WEP مقاله ایی بود که در سال 2001 در مورد مشکلات امنیتی RC4 منتشر شد. در سال 2004 متخصصی بنام Klein مقاله ایی[۵۴] منتشر کرده بود که روش های عنوان شده در حمله قدیمی را بهبود بخشیده و راهکارهایی برای بهینه تر کردن آن، و استفاده از آن علیه WEP ارائه کرده بود. خود این مقاله اگرچه بر روی مشکل RC4 تمرکز کرده بود و امکان استفاده از حمله علیه WEP تنها بصورت یک تئوری در آن مطرح شده بود، اما سه محقق رمزنگاری بنام های Pyshkin, Tews, Weinmann طی تحقیقاتی که در سال 2007 نتایج[۵۵] آنرا منتشر کردند نشان دادند که باز هم روش های بهینه تری برای پیاده سازی تئوری های مطرح شده در تحقیق قبلی (در سال 2004) وجود دارد. جزئیات کار این سه در قالب یک مقاله[۵۶] و ابزاری بنام Aircrack-PTW منتشر شد. کد ابزار منتشر شده توسط این اشخاص بعداً با ابزار Aircrack-ng تلفیق شده و بعنوان یکی از حملات استاندارد در آن پیاده سازی شد. اگر بخاطر داشته باشید در بخش های قبلی همین مقاله عنوان شد که پس از انجام حملاتی برای بدست آوردن IV های لازم، شکستن کلید رمزنگاری WEP با در اختیار داشتن حداقل 40000 پکت داده امکان پذیر است. این امکان و ویژگی در واقع به دلیل پیاده سازی همین حمله PTW در ابزار Aircrack-ng می باشد. این حمله در زمان نگارش این مقاله همچنان آخرین و بهینه ترین روش برای حمله به WEP محسوب می گردد.

استفاده از حمله PTW البته همیشه امکان پذیر نیست. حمله PTW تنها بر اساس تحلیل پکت های ARP قابلیت استخراج و شکستن کلید را دارد و بنا بر این در صورت نیاز به استفاده از این حمله، عمل وادار کردن اکسس پوینت به تولید پکت ها و Weak IV های جدید می بایست حتماً توسط روش های حمله مثل ARP Request Replay انجام گرفته باشد. همچنین حمله PTW تنها در صورتی قابل انجام است که یک تعداد حداقل و مشخص از پکت های داده (از نوع ARP) در دسترس باشد.

ابزار Aircrack-ng برای شکستن WEP در قالب حملات Statistical از تکنیک ها و روش های مختلفی استفاده می کند که در نسخه جاری نرم افزار، در صورت فراهم بودن شرایط، اولین اولویت استفاده از حمله PTW می باشد، و در صورت عدم امکان استفاده از این حمله، روش حمله FMS و KoreK مورد استفاده قرار خواهند گرفت. KoreK Attack در واقع خود شامل 17 تکنیک مختلف حمله می باشد که تمامی آنها در ابزار Aircrack-ng پیاده سازی شده اند. برای آگاهی از جزئیات کامل این ابزار و فراگیری نکات و قابلیت های ابزار برای استفاده بهتر و بهینه تر از آن می توانید به صفحه[۵۷] Wiki آن در وب سایت Aircrack-ng.org مراجعه کنید.

جمع بندی : اگر بخواهیم خلاصه ایی را از مراحل حمله به یک شبکه محافظت شده توسط WEP بصورت عملی بیان کنیم، روال انجام حملات بصورتی که در ادامه گفته می شود خواهد بود.

سناریو حمله به اکسس پوینت های دارای کلاینت : در شرایطی که اکسس پوینت مد نظر ما توسط WEP محافظت شده و در زمان بررسی ما یک یا چند کلاینت به آن متصل و در حال تولید ترافیک باشند روش حمله بطور خلاصه بشرح زیر خواهد بود. با توجه به اینکه در بخش های قبلی مقاله در مورد پارامترها و سوئیچ های مورد استفاده ابزارها توضیحاتی داده شده، در این بخش از تکرار آنها پرهیز شده است.

قدم اول: شروع شنود ترافیک پس از شنایایی و استخراج مک آدرس های کلاینت (ها) و اکسس پوینت، و انتخاب کلاینتی که در حال ارسال پکت های داده (و نه فقط Probe) به اکسس پوینت است. فرض می کنیم که اکسس پوینت مد نظر دارای مک آدرس 11:22:33:44:55:66 و کلاینت انتخاب شده 11:22:33:11:22:33 است.

```
#Airmon-ng start wlan0  6
```

پس از اجرا یک اینترفیس مجازی جدید با نام mon0 و یا Ath0 ایجاد می شود که در حالت monitor mode قرار دارد. کانال مورد استفاده توسط اکسس پوینت قبلاً شناسایی شده (channel 6) بنا بر این در زمان ایجاد، اینترفیس جدید را بر روی این کانال تنظیم کردیم. حال می توان ابزار Airodump را اجرا کرده و آنرا برای تمرکز بر روی اکسس پوینت مورد نظر ما تنظیم کرد :

```
#airodump-ng ath0 –channel 6 –bssid  11:22:33:44:55:66 –a –w dump
```

قدم دوم: انجام حمله Fake Authentication. پیش از اقدام به Packet Injection می بایست کلاینت ما خود را به اکسس پوینت معرفی کند، در غیر اینصورت پکت های ارسالی ما توسط اکسس پوینت Drop خواهند شد.

```
#aireplay-ng -1 0 –q 5 –e net-name –b 11:22:33:44:55:66 ath0
```

قدم سوم : انجام حمله ARP Request Replay. دستور زیر را اجرا کرده و منتظر شوید تا ابزار Aireplay-ng پس از شنود یک پکت ARP مناسب از کلاینت مورد نظر شما که آنرا به ابزار معرفی کرده اید، Packet Injection را آغاز کند:

```
#aireplay-ng -3 –b 11:22:33:44:55:66 –h 11:22:33:11:22:33
```

با شروع Packet Injection تا زمان جمع آوری حداقل 20000 پکت صبر می کنیم. تعداد پکت های داده دریافتی از اکسس پوینت توسط Airodump-ng که قبلاً آنرا اجرا کرده ایم در حال گزارش شدن است.

قدم چهارم : پس از جمع آوری تعداد پکت های لازم، می توان بدون نیاز به متوقف کردن Aireplay-ng اقدام به اجرای Aircrack-ng کرد. در صورتی که کلید پس از طی حداکثر 1 دقیق شکسته نشد، با کمی انتظار برای جمع آوری تعداد بیشتری پکت، ابزار را مجدداً اجرا می کنیم. در صورت وجود تعداد پکت های لازم برای انجام حمله PTW کلید در عرض چند ثانیه شکسته شده و نمایش داده خواهد شد.

```
#aircrack-ng dump.cap
```

سناریو حمله به یک اکسس پوینت بدون کلاینت : در صورتی که با اکسس پوینتی روبرو شویم که هیچ کلاینتی به آن متصل نیست و یا کلاینت دارای هیچ فعالیت ترافیکی قابل استفاده ایی نیست، برای شکستن کلید می بایست از حمله KoreK و یا Fragmentation استفاده کرد. بسته به اکسس پوینت و آسیب پذیر بودن آن شما می بایست هر دو حمله را امتحان کنید:

قدم اول: انجام حمله Fake Authentication. پیش از اقدام به Packet Injection می بایست کلاینت ما خود را به اکسس پوینت معرفی کند، در غیر اینصورت پکت های ارسالی ما توسط اکسس پوینت Drop خواهند شد.

```
#aireplay-ng -1 0 –q 5 –e net-name –b 11:22:33:44:55:66 ath0
```

قدم دوم: انجام حمله KoreK و یا Fragmentation برای بدست آوردن PRGA. بدین منظور از ابزار Aireplay-ng برای حمله KoreK بصورت زیر استفاده می کنیم. با فرض ایکنه شما پیش از حمله مک آدرس سیستم خود را به 11:22:11:22:11:22 جعل کرده اید روال بصورت زیر است. در غیر اینصورت بجای مک آدرس جعلی ذکر شده آدرس واقعی را جایگزین کنید.

```
# aireplay-ng -4 -h 11:22:11:22:11:22 -b 11:22:33:44:55:66 ath0
```

پس از پایان موفقیت آمیز حمله، شما یک فایل با پسوند.xor ویک فایل با پسوند.cap خواهید داشت. در صورت نتیجه بخش نبودن این حمله، می توان از حمله Fragmentation بروش زیر استفاده کرد. در صورت اجرای موفقیت آمیز این حمله، PRGA در فایلی با پسوند.xor ذخیره می گردد.

```
# aireplay-ng -5 -h 11:22:11:22:11:22 -b 11:22:33:44:55:66 ath0
```

قدم سوم : با استفاده از PRGA بدست آمده بوسیله یکی از دو حمله مرحله قبل که در فایل.xor ذخیره شده، ما یک پکت ARP جعلی می سازیم. برای اینکار از ابزار Packetforge بصورت زیر استفاده می شود :

```
#packetforge-ng -0 -h 11:22:11:22:11:22 -a 11:22:33:44:55:66 -k 255.255.255.255 -l 255.255.255.255 -y
SAVED-FILE.xor  -w arp.cap
```

قدم چهارم: برای جمع آوری Weak IV ها آماده می شویم و سپس حمله را شروع می کنیم.

```
#airodump-ng ath0 –channel 6 –bssid  11:22:33:44:55:66 –a –w dump
```

قدم پنجم : در این مرحله، پکت تولید شده در مرحله سوم، مورد استفاده قرار گرفته و بصورت مداوم به اکسس پوینت ارسال می گردد. با شروع Packet Injection تا زمان جمع آوری حداقل 20000 پکت صبر می کنیم. تعداد پکت های داده دریافتی از اکسس پوینت توسط Airodump-ng که قبلاً آنرا اجرا کرده ایم در حال گزارش است.

```
#aireplay-ng -2 -r arp.cap ath0
```

قدم ششم : پس از جمع آوری تعداد پکت های لازم، بدون نیاز به متوقف کردن Aireplay-ng اقدام به اجرای Aircrack-ng می کنیم. در صورتی که کلید پس از طی حداکثر 1 دقیق شکسته نشد، با کمی انتظار برای جمع آوری تعداد بیشتری پکت، ابزار را مجدداً اجرا می کنیم. در صورت وجود تعداد پکت های لازم برای انجام حمله PTW کلید در عرض چند ثانیه شکسته شده و نمایش داده خواهد شد.

```
#aircrack-ng dump.cap
```

حملات سنتی به WPA/WPA2:

با کشف روش های گوناگون برای حمله به WEP، نا امن بودن این پروتکل بارها بصورت عملی به اثبات رسید و با معرفی پروتکل WPA (WiFi Protected Access) استفاده از آن بعنوان جایگزینی امن و مناسب پیشنهاد شد. WPA بسیاری از نقاط ضعفی که در WEP از آنها برای حمله سو استفاده شده را مرتفع کرده و پیاده سازی برخی از حملات موثر که از آنها در بخش حملات مدرن یاد شد را عملاً غیر ممکن و یا بسیار مشکل کرده است. اگر چه WPA جایگزین WEP شده و بر اساس مکانیزم های کاری پیاده سازی شده و مورد استفاده در این پروتکل امکان انجام حملات همانند آنچه در WEP شاهد آن بودیم وجود ندارد، اما در ادامه خواهیم دید که بدلیل اشتباه در پیاده سازی کامل و صحیح برخی موارد، حتی WPA نیز می تواند آسیب پذیر باشد. برای درک اینکه چرا حملات WEP بر روی WPA کارساز نیستند، و اینکه ابزارها و تکنیک های حمله موجود علیه WPA چگونه عمل می کنند آگاهی از مکانیزم کاری WPA و تغییراتی که در مقایسه با WEP در آن اعمال شده الزامیست.

مکانیزم کاری WPA : اساس کاری پروتکل امنیتی WPA در واقع بسیار شبیه به WEP می باشد با این تفاوت که با اضافه کردن چندین راهکار امنیتی در لایه های مختلف سعی در امن سازی این پروتکل نسبت به حملات رایج موجود برای WEP شده است. اساس رمزنگاری و امنیت در WPA همچنان مبتنی برRC4 الگوریتم بوده و در WPA نیز همانگونه که در بخش WEP گفته شد از IV در کنار کلید رمزنگاری اصلی برای انجام چرخه رمزنگاری استفاده شده است. روش کنترل Integrity پکت ها در WPA همچنان مبتنی بر نوعی امن تر از CRC32 می باشد (الگوریتم Michael[۵۸]) که یک مکانیزم امنیتی نیز به آن اضافه شده. همه موارد ذکر شده در WPAدارای پیاده سازی بهتر و کامل تری نسبت به WEP می باشند. بدین معنی که سعی شده نقاط ضعف هر یک از بخش های یاد شده از WEP که مورد حمله قرار گرفته بود توسط روش هایی بهبود داده شده و مورد استفاده قرار گیرد. مجموعه این تغییرات و موارد بهبود یافته در قالب یک پروتکل امنیتی جدید بنام TKIP[۵۹] (Temporal Key Integrity Protocol)عرضه و بعنوان مبنای کاری WPA مورد استفاده قرار گرفت. اگر بخواهیم بصورت خلاصه موارد تغییر یافته توسط TKIP را مرور کنیم می توان آنها را در قالب سه تغییر کلی و اصلی معرفی کرد.

در اولین مورد بر خلاف WEP که از یک رشته 24 بیتی بعنوان IV استفاده می نماید، TKIP در WPA از یک رشته 48 بیتی برای این منظور استفاده می کند. پس مشکل کوچک بودن دامنه اعداد تصادفی تولید شده بعنوان IV تا حد زیادی پوشش داده شده. علاوه بر این در TKIP بر خلاف WEP، از کلید رمزنگاری بصورت مستقیم برای تولید Master Key مورد استفاده در پروسه رمزنگاری و الگوریتم RC4 استفاده نمی شود بلکه کلید رمزنگاری اولیه با یک رشته تصادفی 64 بیتی ترکیب شده و از حاصل این ترکیب کلید نهایی رمزنگاری بدست می آید. روال تولید رشته تصادفی و تکمیل پروسه رمزنگاری برای تولید کلید 256 بیتی نهایی در TKIP خود به نوعی یک مکانیزم امنیتی کمکی برای TKIP محسوب می شود زیرا الگوریتم مورد استفاده بدین منظور از نظر پردازشی و محاسباتی یک الگوریتم بسیار سنگین و زمانبر می باشد. بمنظور تولید کلید رمزنگاری در TKIP، پس از ترکیب رشته رمز (با طول بین 8 تا 63 کاراکتر) که کاربر مشخص کرده با رشته مشخص شده بعنوان SSID، مقادیر ذکر شده بعنوان ورودی یک الگوریتم بنام HMAC-SHA1 استفاده می شوند و این چرخه تولید رشته نهایی 4096 بار تکرار می شود تا کلید نهایی محاسبه گردد. ای عمل در واقع شبیه 4096 بار محاسبه SHA1 Hash از یک عبارت ترکیب شده با Seed است. مجموعه این عملیات که در طی آن کلید نهایی رمزنگاری WPA تولید می شود تحت عنوان پروتکل BBPKDF2 شناخته می شود. پروتکل BBPKDF2 در ابتدای هر اتصال (Association & Authorization) اجرا شده و پس از انجام آن در طول ارتباط از کلید 256 بیتی حاصل از آن برای رمزنگاری استفاده می شود. از این کلید رمزنگاری تولید شده بنام PMK (Pairwise Master Key) نیز یاد می شود. نکته دیگر اینکه در TKIP بدلیل مکانیزم خاص پیاده سازی شده کلید رمزنگاری (PMK) پس از ترکیب با یک سری

متغیرهای دیگر (مانند IV و Mac Address های دو سوی ارتباط) مورد استفاده قرار می گیرد. به این کلید رمزنگاری ثانویه و نهایی تولید شده PTK (Pairwise Transient Key)گفته می شود. بکمک این مکانیزم در عمل محتوای رمزنگاری شده دو داده یکسان توسط یک کلید نیز متفاوت خواهند بود. همچنین کلید رمزنگاری در هر نشست (به ازای هر کلاینت) نیز بدلیل متفاوت بودن پارامترهای استفاده شده در تولید PMK متفاوت خواهد بود. این خاصیت سبب می شود تا هیچ یک از حملات رمزنگاری Related Key Attack که بر روی WEPقابل استفاده بود عملاً کارساز نباشند.

تغییر دوم در WPA که در TKIP پیاده سازی شده راهکار مقابله با حملات علیه Integrity پکت است، بطوری که در TKIP در صورتی که در یک نشست دو بار خطایی در Integrity پکت های ارسالی مشاهده شود هم اکسس پوینت و هم کلاینت بمدت 60 ثانیه ارسال و دریافت پکت ها را متوقف کرده و همچنین اکسس پوینت اقدام به تولید یک کلید رمزنگاری جدید (Re-Keying) برای ادامه ارتباط می نماید. این مکانیزم پیاده سازی حملاتی مانند Chop-Chop را که علیه WEP کارساز بود بسیار مشکل می کند زیرا با بروز خطای دوم، کلید رمزنگاری توسط اکسس پوینت تغییر کرده و نفوذگر می بایست حمله خود را مجدداً از سر گیرد، اگر چه خود این مکانیزم امنیتی در TKIP به نوعی راهی را برای پیاده سازی حملات DoS باز کرده است.

در نهایت سومین تغییر اساسی و مکانیزم امنیتی که در TKIP پیاده سازی شده، بمنظور مقابله با حملات Request Replay است. در این مکانیزم با استفاده از نظارت و کنترل ترتیب (Sequence) پکت های ارسالی، در صورت مشاهده پکت های نا متعارف این پکت ها از سوی اکسس پوینت reject می شوند. این یعنی اینکه شما دیگر قادر به تولید و تزریق تعداد نا محدودی پکت به اکسس پوینت و وادار کردن آن به باز تولید آنها (ARP Request Replay Attack in WEP) نخواهید بود. در WEP بدلیل عدم کنترل Sequence Number پکت های ارسالی برای تکراری ویا نا مرتب بودن روال ارسال آنها، انجام این روش حمله میسر بود.

با توجه به اینکه نسخه اولیه WPA که مبتنی بر TKIP بود همچنان از الگوریتم های RC4 برای رمزنگاری و CRC32 برای کنترل صحت داده ها استفاده می کرد، پروتکل جدیدی بنام CCMP معرفی گردید که بر خلاف TKIP الگوریتم رمزنگاری AES جایگزین RC4 و CRC32 گردید. CCMP کلیه ضعف های امنیتی و حملات شناخته شده علیه WEP و TKIP را پوشش داده و در حال حاضر بعنوان تنها گزینه قابل اطمینان مطرح می باشد. علت عدم ارائه CCMP در زمان تولد WPA شاید نیازمندی های سخت افزاری این پروتکل می باشد. استفاده از WPA بر روی اکسس پوینت ها و کارت های بیسیم که پیش از این فقط از WEP پشتیبانی می کردند تنها با یک بروز رسانی نرم افزاری (Firmware Upgrade) ممکن می شد. اما بدلیل استفاده از AES در پروتکل CCMP، علاوه بر بروز رسانی نرم افزاری استفاده از سخت افزارهای بروز شده نیز الزامیست و قابلیت استفاده از AES می بایست در سخت افزار و چیپ اکسس پوینت و کارت شبکه بیسیم گنجانده شود. به همین دلیل ابتدا TKIP معرفی شده و به مرور از سال 2003 به بعد تولید کننده گان سخت افزار پشتیبانی از AES و CCMP را بعنوان استاندارد در محصولات خود گنجاندند.

WPA2 : WPA2 نیز از اساسی مشابه WPA استفاده می کند با این تفاوت که برخی قابلیت های جدید و امن تر برای مدیریت و کنترل کلید ها و همچنین اعتبارسنجی به آن اضافه شده است. CCMP و TKIP هر دو در WPA2نیز پشتیبانی می گردند. اغلب تغییرات اعمال شده در WPA2 در مقایسه با WPA مربوط به افزایش کارایی و قابلیت های Mobility پروتکل بوده است و به همین دلیل از بحث در مورد آنها در این مقاله صرف نظر گردیده است. نکته اصلی و تغییر مهم در WPA2 که موجب ارتقاع امنیت در آن می شود، پروتکل CCMP می باشد که در بخش قبل به آن اشاره شد. در نظر داشتن این نکته الزامیست که مدت تضمین اعتبار و امنیت WPA در زمان استاندارد شدن آن (سال 2004) تنها 5 سال اعلام گردید و WPA/TKIP در واقع تنها یک فرجه زمانی برای فراهم شدن شرایط لازم در شبکه های بزرگ و پیاده سازی پروتکل و مکانیزم های امنیتی معرفی شده در استاندارد 802.11i[۶] بوده است. اصلی ترین مکانیزم امنیتی معرفی شده در

802.11i در واقع پروتکل امنیتی اعتبارسنجی 802.1x[۶۱] می باشد. 802.1x روشی بسیار کارامد و امن تر را در مقایسه با TKIPو یا CCMP در حالت کاری معمولی (Personal Mode) برای مدیریت کلید های رمزنگاری فراهم می آورد. اگرچه استفاده از پروتکل 802.1x سطح امنیت را در شبکه های بیسیم بسیار افزایش می دهد اما خود این مکانیزم نیز دارای ضعف های امنیتی خاص می باشد که امکان حمله و سو استفاده از آنرا فراهم می آورد. پوشش این حملات و تشریح تکنیک های حمله به یک شبکه محافظت شده توسط 802.1x خارج از حوصله این متن می باشد.

حال که بخشی از جزئیات کاری پروتکل های WPA/2 مورد بررسی قرار گرفت می توان با دید بهتری حملات موجود علیه آنرا مورد بررسی قرار داد و علت کارآمد بودن یا ناکارآمد بودن برخی حملات را شناخت.

حملات Dictionary علیه WPA/2 : گفته شد که بدلیل تغییرات عمده اعمال شده در WPA توسط TKIP، بسیاری از نکات و نقاط ضعفی که از آنها برای حمله استفاده می شد پوشش داده شده و انجام این حملات را در شرایط واقعی بسیار مشکل و در برخی موارد کاملاً نا ممکن کرد. همانند زمانی که حمله موثری برای WEP کشف نشده بود، تا سال 2008 حملات سنتی از نوع Dictionary Attack تنها راه معرفی شده برای بدست آوردن کلید رمزنگاری WPA/2 شناخته می شد. اگر چه از سال 2008 تا کنون تحقیقات جدیدی در مورد مشکلات امنیتی WPA/TKIP انجام و منتشر شده است اما این تکنیک های حمله هنوز در مراحل اولیه خود بوده و آنطور که باید و مشابه WEP، بصورت عملی (Practical) کاربردی ندارند.

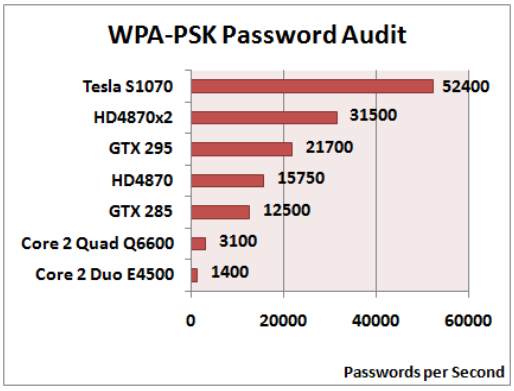
پیش از این عنوان شد که بر خلاف WEP که در آن کلیه پکت ها بصورت یکسان و با یک کلید مشخص رمزنگاری می شوند در WPA به ازای هر کلاینت و حتی هر پکت متغیر های مورد استفاده در رمزنگاری تغییر می کند. بنا بر این مشابه آنچه در مورد WEP صادق بود نمی توان به صرف در اختیار داشتن یک پکت رمزنگاری شده اقدام به کشف کلید رمزنگاری کرد، زیرا حتی در صورت موفقیت آمیز بودن حمله نیز شما تازه کلید رمزنگاری PTK را بدست آورده اید که خود این کلید از کلید رمزنگاری اصلی (PMK) مشتق شده است. جدای از این موضوع تعداد متغیرهایی که برای محاسبه PTK لازم است بسیار بیشتر از PMK بوده و در نتیجه روال انجام حمله نیز پیچیده تر می شود. بهترین راهی که برای بدست آوردن مقدار PMK بنظر می رسد، شنود شبکه و بدست آوردن PMK از ترافیک عبوری است. اگر بخاطر داشته باشید گفته شد که PMK در هر نشست تنها یک بار تولید شده و پس از آن تنها کلید های مشتق شده از آن (PTK) در طول ارتباط مورد استفاده قرار می گیرند. تولید PMK در زمان شروع برقراری نشست در Authentication و انجام 4 way Handshake کلاینت با اکسس پوینت اتفاق می افتد، بنا بر این شرط بدست آوردن PMK (و امکان انجام Dictionary Attack) این است که شما بتوانید در لحظه شروع نشست و انجام 4 way Handshake ترافیک بین کلاینت و اکسس پوینت را شنود کنید. عمل Authentication در WPA/2 در قالب ارسال و دریافت چند پکت اول ارتباط انجام می شود و شنود این پکت ها (که عملاً 4 پکت مهم هستند) مساویست با بدست آوردن PMK. پس از این روال حمله مشخص است.

با توجه به اینکه در الگوریتم مورد استفاده برای تولید PMK تنها یک متغیر اصلی (بجز خود کلید اصلی مشخص شده توسط کاربر) وجود دارد و این متغیر نیز همان SSID شبکه است، می توان با تکرار این الگوریتم و جایگزین کردن کلمات دیکشنری در روال محاسبه اقدام به حدس زدن کلید رمز تعریف شده توسط کاربر نمود. بطور ساده فرمولی که از آن برای تولید کلید استفاده می شود مشابه این عبارت است که:

PMK = PBKDF2 (user defined pass, SSID, 4096, 256)

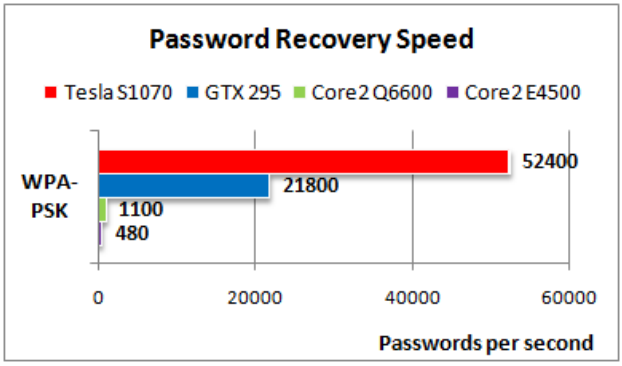
با وجود آگاهی از SSID، در صورتی که نتیجه نهایی بدست آمده از این فرمول با مقدار PMK که پیش از این شنود شده یکسان شود، ما کلمه رمز استفاده شده توسط کاربر را کشف کرده ایم. اما در این حمله یک مشکل اساسی وجود دارد و آن هم سنگینی بار محاسباتی این فرمول است. در PBKDF2 برای محاسبه کلید 256 بیتی نهایی پس از ترکیب SSID و کلمه رمز کاربر، عبارت بدست آمده 4096 بار در فرمول تولید هش SHA-1 قرار داده می شود تا عبارت نهایی حاصل شود. حتی بار پردازشی یک بار محاسبه SHA-1 هم در مقایسه با بسیاری از الگوریتم های دیگر سنگین است چه رسد به 4096 بار تکرار آن ! این مورد سبب شده است تا حتی حملات نوع Dicrionary Attack نیز علی رغم امکان پیاده سازی با کندی زیاد صورت پذیرند بطوری که حتی در یک سیستم دارای پردازنده بسیار سریع امروزی تعداد PMK های محاسبه شده از چند هزار مورد در ثانیه تجاوز نکند. سرعت محاسبه چند هزار کلید در ثانیه به نوعی برای سیستم ها و نرم افزارهای عادی حمله به WPA یک سرعت بسیار بالا و ایده آل محسوب می شود. در یک سیستم با پردازنده معمولی و نرم افزار بهینه نشده این عدد به کمتر از یک هزار تلاش در ثانیه می رسد. برای مقایسه و درک میزان کندی این محاسبه، سرعت ذکر شده را با سرعت محاسبه چند میلیون کلید در ثانیه برای الگوریتمی مانند RC4 مقایسه کنید. کندی سرعت سبب شده گرایش به سمت استفاده از تکنولوژی های پردازشی سریع تر مانند FPGA و پردازنده های گرافیکی (GPU) شده است.

از جمله شناخته شده ترین نرم افزارهایی که از Dictionary Attack علیه WPA/2 پشتیبانی می کنند نرم افزار Aircrack-ng و CoWPAtty می باشند. در هر دو مورد اشاره شده نمونه هایی که از FPGA و GPU (تکنولوژی CUDA شرکت Nvidia) پشتیبانی می کنند نیز پیاده سازی شده اند که در بخش حملات سنتی به WEP به آنها اشاره شد. بعنوان مثال نسخه مبتنی بر CUDA از نرم افزار Aircrack-ng از طریق SVN این ابزار و در یک شاخه اختصاصی[۶۲] در دسترس است. ابزار مشهور CAIN یکی از نمونه های آزاد در دسترس برای انجام حمله در محیط سیستم عامل ویندوز می باشد. یکی دیگر از نرم افزارهای مناسب و بهینه (از نظر سرعت پردازش) بسته نرم افزاری ارائه شده توسط شرکت ElecomSoft می باشد که تحت نام "Wireless Security Auditor"[۶۳] بصورت تجاری عرضه شده است. این نرم افزار جزو معدود ابزارهای مورد استفاده برای Dictionary Attack علیه WPA است که از GPU برای افزایش سرعت پشتیبانی می کند. بدلیل هزینه بالای استفاده از تکنولوژی FPGA بسیاری از کاربران عادی قادر به تهیه سخت افزارهای لازم نمی باشند اما هر کامپیوتر مدرن امروزی با هزینه ایی معقول می تواند به یک GPU پر سرعت مجهز گردد. شکل زیر که از وب سایت تولید کننده نرم افزار برگرفته شده مقایسه ایی از سرعت محاسباتی این ابزار بر روی بستر های پردازشی مختلف را نشان می هد.. در سیستم مورد استفاده نویسنده این نرم افزار قادر به انجام حمله با سرعت متوسط 3900 رمز در ثانیه تنها با استفاده از پردازنده اصلی (Intel i7 920) و سرعت متوسط 5380 رمز در ثانیه در حالت استفاده از GPU (Nvidia 9500GT 1300Mhz) در کنار پردازنده اصلی می باشد.



استفاده از قدرت پردازشی چندین کامپیوتر بصورت موازی یکی دیگر از روش های بهبود سرعت حمله می باشد. در این مورد نیز شرکت ElecomSoft به نوعی پیشتاز بوده و بسته نرم افزاری Distributed Password Recovery آن شاید تنها نمونه ابزار پایدار در دسترس

(چه بصورت تجاری و چه آزاد) برای حمله به WPA در قالب Dictionary Attack بصورت Distributed باشد. ترکیب امکان Distributed Attack با قابلیت استفاده از GPU در هر سیستم می تواند بصورت چشمگیری سرعت حمله را افزایش دهد. در نمودار مقایسه این نرم افزار که در زیر آورده شده است، سرعت محاسباتی برای یک پردازنده خاص (Core 2 E4500) تقریباً یک سوم سرعت اعلام شده در نرم افزار قبلی همین شرکت یعنی "Wireless Security Auditor" می باشد. بدلیل عدم آزمایش این مورد توسط نویسنده این مقاله دقیقاً مشخص نیست که آیا الگوریتم حمله پیاده سازی شده در WSA سریع تر و بهینه تر از نمونه Distributed می باشد ویا این تنها یک اشتباه تبلیغاتی و یا شاید تفاوت در نسخه های مورد استفاده برای بررسی می باشد.



گفته شد که بمنظور شکستن کلید WPA در اختیار داشتن پکت های رد و بدل شده در زمان Authentication الزامیست. بمنظور شنود این پکت ها می توان از ابزار Airodump-ng استفاده نمود. در صورتی که در زمان شنود توسط این ابزار (یا هر نرم افزار شنود دیگر که خروجی با فرمت PCap ارائه می دهد) کلاینتی Authenticate شده و به اکسس پوینت متصل شود شما به آنچه نیاز داشتید دست یافته اید. اما در صورتی که شما در آن لحظه در حال شنود نباشید راهکار چیست؟ در این صورت می توان از تکنیک حمله Deauthentication علیه یکی از کلاینت های فعال شبکه استفاده کرد. در این حالت نفوذگر با قطع کردن تعمدی ارتباط بین کلاینت و اکسس پوینت او را مجبور به Authentication مجدد می کند. حمله Deauthentication را می توان توسط ابزار Aireplay-ng که پیش از این نیز مورد استفاده قرار گرفت، انجام داد. Aireplay این امکان را فراهم می کند که کلیه کلاینت های یک اکسس پوینت را مورد حمله قرار داده و یا یک کلاینت خاص بدین صورت مورد حمله واقع شود. روش استفاده از این ابزار بصورتی که در زیر آورده شده است. بمنظور شنود PMK کافیست پیش از اجرای Aireplay-ng ابزار Airodump-ng یا هر sniffer دیگر برای شنود از یک اینترفیس که در حالت Monitor Mode قرار دارد اجرا شده باشد.

```
aireplay-ng -0 5 -a 00:11:22:33:44:55 -c 00:11:00:22:00:33 ath0
```

در این حالت، "0-" معرف حمله Deauthentication بوده و عدد "5" تعداد پیغام های .Deauth است که فرستاده خواهد شد. "a-" معرف bssid (مک آدرس اکسس پوینت) و "c-" نیز معرف مک آدرس کلاینت می باشد. در صورت مشخص نکردن کلاینتی با سوئیچ "c-" ابزار بصورت Broadcast سعی در .DeAuth کردن کلیه کلاینت های متصل خواهد نمود. توجه به این نکته ضروریست که موفقیت این حمله منوط به این است که کلاینت مورد حمله واقع شده در محدوده پوشش کارت شبکه بیسیم شما قرار داشته و بتواند پکت های ارسالی شما را دریافت کند. بمنظور شنود نیز توصیه می شود با استفاده از فیلتر های تعریف شده در ابزار Airodump شنود و ذخیره ترافیک دریافتی محدود به همان اکسس پوینتی شده باشد که شما قصد حمله به آنرا دارید. بدین منظور پس از قرار دادن اینترفیس در حالت Monitor Mode می توان ابزار را بروش زیر مورد استفاده قرار داد :

```
airodump-ng -c 6 --bssid 00:11:22:33:44:55-w out-file.pcap ath0
```

سوئیچ "6-" ابزار را بر روی کانال 6 قفل کرده و توسط پارامتر داده شده به سوئیچ "bssid—" نرم افزار در زمان اجرا تنها پکت های مربوط به اکسس پوینت مشخص شده را نمایش داده و ذخیره می کند. پس از اتصال مجدد کلاینت هدف قرار گرفته به اکسس پوینت، فایل.pcap ذخیره شده را می توان به عنوان ورودی برای ابزارهایی مانند Aircrack-ng یا coWPAtty مورد استفاده قرار داد. در وصورت استفاده از Aircrack-ng برای انجام Dictionary Attack روال اجرای ابزار بصورت زیر خواهد بود.

```
Aircrack-ng –w dictionary-file.txt out.pcap
```

نرم افزار Aircrack-ng پس از کنترل فایل pcap ورودی خود bssid مورد نیاز را استخراج کرده و حمله را آغاز می کند. در صورتی که handshake های ذخیره شده بیشتر از یک bssid در فایل ورودی وجود داشته باشد، ابزار طی پیغامی از کاربر می خواهد که bssid مورد نظر خود را انتخاب نماید. این قابلیت در ابزارهایی مانند coWPAtty وجود نداشته و کاربر توسط سوئیچ های مشخص، می بایست ESSID را به نرم افزار معرفی کند. در ابزار aircrack-ng در صورتی که به هر دلیل امکان تشخیص ESSID از طریق تحلیل پکت ها وجود نداشته باشد، کاربر توسط سوئیچ اضافی "e- ESSID" می بایست آنرا برای نرم افزار مشخص کند. نتیجه یک حمله موفقیت آمیز مشابه تصویر زیر می باشد.

```
unknown temp # aircrack-ng -w /pentest/dic/wordlist.txt /root/wpa.pcap
Opening /root/wpa.pcap
Read 1093 packets.

   #  BSSID              ESSID                     Encryption

   1  00:0C:41:82:B2:55  Coherer                   WPA (1 handshake)
   2  FF:FF:FF:FF:FF:3F                            WEP (1 IVs)
   3  81:F8:47:33:56:BB                            Unknown
   4  98:D3:04:64:FA:55                            WPA (0 handshake)
   5  65:78:F7:B7:30:84                            Unknown
   6  F4:9F:8F:EA:7B:E6                            Unknown
   7  65:78:F7:B7:60:A9                            Unknown
   8  92:F3:65:74:D2:DB                            Unknown

Index number of target network ? 1
```

```
                              Aircrack-ng 1.0 rc4 r1623

                    [00:00:01] 4600 keys tested (2426.56 k/s)

                              KEY FOUND! [ Induction ]


      Master Key     : A2 88 FC F0 CA AA CD A9 A9 F5 86 33 FF 35 E8 99
                       2A 01 D9 C1 0B A5 E0 2E FD F8 CB 5D 73 0C E7 BC

      Transient Key  : B1 CD 79 27 16 76 29 03 F7 23 42 4C D7 D1 65 11
                       82 A6 44 13 3B FA 4E 0B 75 D9 6D 23 08 35 84 33
                       15 79 8D 51 1B EA E0 02 83 13 C8 AB 32 F1 2C 7E
                       CB 71 C8 93 48 26 69 DA AF 0E 92 23 FE 1C 0A ED

      EAPOL HMAC     : A4 62 A7 02 9A D5 BA 30 B6 AF 0D F3 91 98 8E 45
unknown ~ #
```

حملات Pre-Computed علیه WPA/2 : بدلیل کندی پروسه شکستن رمز WPA/2 برخی به فکر استفاده از روش های حمله مبتنی بر Pre-Computer Tables افتاده و ابزارهایی را بدین منظور تولید کردند. استفاده از تکنیک پیاده سازی Rainbow Table[۶۴] برای الگوریتم های مختلف سالهاست که رواج داشته و در برخی موارد مانند NTLM و یا MD5 علی رغم ارتقاع سرعت و قدرت پردازنده ها هنوز هم کارآمد و دارای سرعت قابل قبولی در بازیابی کلمات عبور از مقدار Hash آنها می باشد. اصولاً هرچه روند تولید Hash کند تر و از نظر محاسباتی سنگین تر باشد، استفاده از Rainbow Table ها برای شکستن آنها مقرون به صرفه تر خواهد بود. این در مورد WPA نیز صدق می کند. سرعت در روش Dictionary Attack قابل مقایسه با روش Rainbow (pre-computed) table نبوده و بسیار کند تر است بطوری که سرعت حمله بدین روش بر روی یک سیستم با پردازنده معمولی با سرعت حمله در یک سیستم مجهز به سریع ترین GPU های موجود نزدیک است.

اساس کار Rainbow Table ها بسیار ساده است. بجای اینکه به ازای هر Hash یکبار کلیه کلمات عبور ممکن تولید و تست شوند، لیستی از کلمات عبور رایج تهیه شده و Hash مربوط به هر مورد محاسبه شده، در یک Table ذخیره می گردد. حال در صورت نیاز به بدست آوردن مقدار Clear-text یک Hash کافیست Table مذکور برای وجود Hash مورد نظر جستجو شده و مقدار clear-text آن استخراج گردد. از نظر سرعت و بار پردازشی جستجو در حافظه بسیار سریعتر از تکرار پروسه تولید Hash و مقایسه نتیجه است. برای تولید یک Table اگرچه زمان زیاد و فضای ذخیره سازی قابل توجهی مورد نیاز است اما این عمل تنها یکبار انجام می شود و پس از آن تنها کار، جستجوی Table های تولید شده برای مقدار Hash خواهد بود. در بسیاری از سیستم ها البته راهکاری برای مقابله با این تکنیک حمله وجود دارد. استفاده از Salt در تولید Hash براحتی این تکنیک حمله را بر بسیاری از موارد نا کارآمد خواهد کرد. Salt یک مقدار تصادفی است که قبل از تولید Hash و در مواردی بعد از آن در کنار کلمه عبور اصلی (یا مقدار Hash شده آن) قرار می گیرد و باعث می شود تا کلمه های عبور یکسان، مقدار Hash متفاوتی داشته باشند. بدین ترتیب برای تولید Rainbow Table برای کلمات عبور، به ازای هر مقدار Salt بکار رفته می بایست یکسری Table جدید تولید گردد و این یعنی حجمی نجومی از فضای دیسک مورد نیاز و سالها پردازش برای تولید.

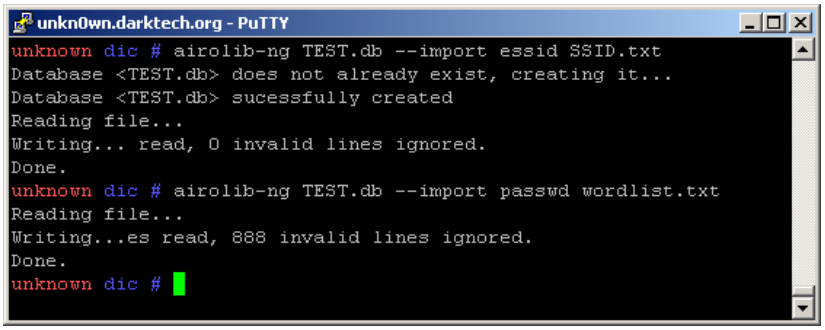
از این تکنیک امن سازی در تولید EAPOL HMAC Hash (یا همان Hash نهایی مد نظر ما برای حمله) در WPA نیز استفاده شده است. در پروسه تولید EAPOL HMAC که پیش از این در مقاله بنام PMK معرفی شد، از SSID بعنوان مقدار Salt استفاده می شود. استفاده از SSID بعنوان Salt بدین معنی است که برای استفاده از روش حمله مبتنی بر Rainbow Table، به ازای هر SSID می بایست یک سری Table تولید شود. این مورد اگرچه کاربرد این تکنیک حمله را محدود کرده است اما شاید دانستن این نکته که بسیاری از کاربران از SSID های رایج و مشابه در تنظیمات شبکه خود استفاده می کنند، کمی امیدوار کننده باشد. در صورتی که شما قصد تولید اینTable ها را دارید می توانید از لیست های SSID های رایج[۶۵] برای تولید استفاده کنید. در صورتی که قصد تولید یک Table کارامد را دارید، علاوه بر استفاده از لیست های موجود از SSID های رایج، می توانید خود نیز پس از انجام War-driving لیستی از محبوب ترین نام های استفاده شده در محدوده خود را به موارد موجود اضافه کنید.

نرم افزارهای مختلفی این روش حمله را در کنار امکانات خود ارائه کرده اند. CoWPAtty و Aircrack-ng هر دو بخوبی این تکنیک را پیاده سازی کرده اند. مزیت پیاده سازی نرم افزار Aircrack-ng پشتیبانی از فرمت Table های استاندارد و همچنین فرمت اختصاصی نرم افزار CoWPAtty می باشد و اطلاعات در بانک اطلاعاتی استاندارد SQLite ذخیره سازی می گردد. همچنین بروز رسانی و کامل کردن Table ها در بسته نرم افزار Aircrack-ng بسیار آسان تر از CoWPAtty است. در بسته نرم افزاری Aircrack، ابزار Airolib-ng بمنظور ایجاد Table ها مورد استفاده قرار می گیرد، و پس از ساخته شدن این Table ها توسط خود نرم افزار Aircrack-ng مورد استفاده قرار می گیرد. بدلیل بهینه تر بودن پیاده سازی این تکنیک در بسته Aircrack-ng مراحل استفاده و حمله مربوط به این ابزار مرور خواهد شد.

برای تهیه Table ها و استفاده از این روش ابتدا لیستی از SSID های رایج و لیستی از کلمات عبور رایج را بصورت دو فایل متنی مجزا تهیه می کنیم. سپس با استفاده از ابزار Airolib-ng مطابق روش زیر، آنها را در Table درج می کنیم :

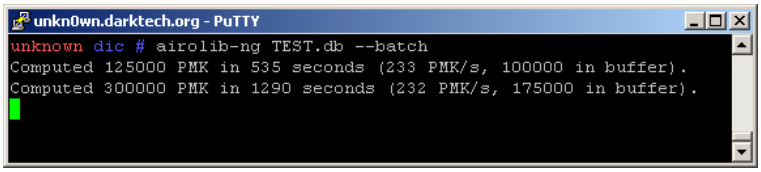
```
Airolib-ng TEST.db –import essid top-100-ssid.txt
Airolib-ng TEST.db –import passwd wordlist.txt
```

در صورتی که Table مذکور وجود نداشته باشد ابزار آنرا بصورت خودکار ایجاد می کند و اگر فایلی با این نام وجود داشته باشد اطلاعات جدید در آن درج می گردند.



پس از آماده شدن جدول، می بایست شروع به تولید PMK برای هر یک از essid های درج شده در آن نمود. Airolib-ng به ازای هر essid موجود، PMK کلیه کلمات عبور درج شده را محاسبه و ذخیره می نماید. این عمل بسیار زمانبر بوده و در صورتی که تعداد کلمات عبور درج شده زیاد باشد ممکن است ساعت ها و حتی روزها بطول بیانجامد. بروش زیر محاسبه توسط ابزار شروع می گردد:

```
Airolib-ng TEST.db --batch
```

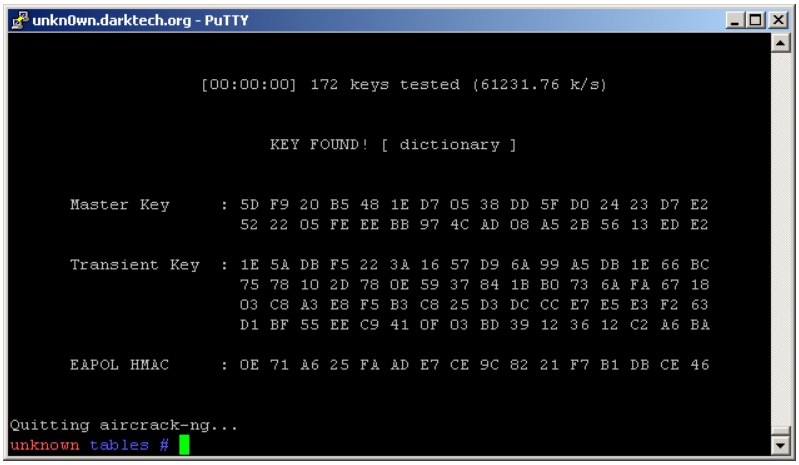


با توجه به طولانی بودن مدت انجام این مرحله، در صورت نیاز به آگاهی از اینکه محاسبه چه تعدادی از essid های داده شده باقی مانده می توانید بدون متوقف کردن ابزار Airolib-ng، در خط فرمان جدیدی از دستور زیر استفاده کنید :

```
Airolib-ng TEST.db –stat |grep –iv 100 | wc -l
```

خواهید دید که سرعت ثبت PMK های تولید شده در جدول بسیار کمتر از زمانی است که ابزار Aircrack-ng در حال حدس کلمات عبور بصورت معمولیست. دلیل این موضوع دخیل شدن مرحله خواندن/نوشتن بر روی دیسک سخت و درج در بانک اطلاعاتی ساخته شده است. اما این کندی اولیه ارزش وقت صرف شده را دارد. با اتمام این مرحله و کامل شدن بانک اطلاعاتی می توان از ابزار Aircrack-ng برای بکار گرفتن آن استفاده کرد. کافیست بجای مشخص کردن فایل دیکشنری همانند آنچه در روش Dictionary-Attack انجام داده شد، فایل بانک اطلاعاتی بصورت زیر فراخوانی شود :

```
Aircrack-ng –r TEST.db wpa-capture.cap
```



به سرعت ابزار در زمان شکستن رمز دقت کنید و آنرا با سرعت حمله در روش Dictionary-Attack مقایسه کنید. خواهید دید که سرعت در این روش در حدود 30 برار افزایش یافته است ! توجه داشته باشید که این سرعت بالا با استفاده از CPU خود سیستم بدست آمده است و در روش Dictionary-Attack حتی اگر از GPU استفاده شود، تنها با استفاده از کارت های گرافیکی بسیار گران قیمت و یا کارت های جانبی TESLA شرکت Nvidia که بیش از 1200 دلار قیمت دارند می توان به چنین سرعتی دست یافت. تنها اشکال این روش این است که بانک اطلاعاتی شما می بایست SSID هدف شما را پوشش داده و همچنین فایل دیکشنری کلمات عبور می بایست بسیار کامل باشد، زیرا در این روش پس از تهیه بانک اطلاعاتی اولیه، اضافه کردن حتی یک کلمه عبور جدید به جدول و محاسبه PMK آن به ازای هر SSID ثبت شده زمانبر خواهد بود. روش های مختلفی نیز برای افزایش راندمان و سرعت در خلال تولید بانک اطلاعاتی وجود دارد که از آن جمله می توان به تقسیم اطلاعات بین بانک های اطلاعاتی مختلف و اجرای موازی Airolib-ng اشاره کرد، و یا استفاده از FPGA و یا GPU در روند تولید جدول. ترکیب روش های ذکر شده و پیاده سازی آن بصورت Distributed و بین چند کامپیوتر راهکار دیگری است که می توان از آن استفاده کرد. در نهایت سریع ترین ترکیب ممکن و آزمایش شده توسط نویسنده استفاده از RAM-Drive بعنوان بستر ذخیره سازی بانک اطلاعاتی در کنار بهره گیری از تکنولوژی GPU می باشد. Pyrit[۶۶] یکی از ابزارهای کارآمد دیگر برای حمله به WPA/2 است که علاوه بر اینکه از تکنیک Rainbow Table پشتیبانی می کند، می توان با استفاده از آن روش های ذکر شده برای افزایش راندمان و سرعت حمله را نیز پیاده سازی کرد. نقطه قوت این نرم افزار قابلیت بهره گیری از GPU برای محاسبات و همچنین پشتیبانی از سیستم های بانک اطلاعاتی مختلف مانند MySQL و یا MS-SQL است. نحوه استفاده از این ابزار ساده بوده و اساس کار آن مشابه Airolib-ng می باشد. به همین دلیل از توضیحات بیشتر در مورد این ابزار صرف نظر شده و بررسی و فراگیری روش استفاده ازآن به عهده خواننده مطلب گذاشته می شود.

در صورتی که شما پهنای باند زیادی برای دسترسی به اینترنت در اختیار دارید، می توانید از جدول های از پیش تهیه شده استفاده کرده و آنها را دانلود کنید. اغلب جداول در دسترس به فرمت نرم افزار CoWPAtty منتشر شده اند اما با توجه به اینکه Airolib-ng می تواند این جداول را مستقیماً به فرمت استاندارد خود تبدیل کند، شما مشکلی نخواهید داشت. دو نمونه از این جداول آماده توسط Offensive-Security[67] و Wifi-Church[۶۸] منتشر شده و در دسترس عموم قرار دارند.

حملات پیشرفته به WPA :

روش هایی که تا اینجا برای حمله به WPA/2 معرفی شده و مورد بحث قرار گرفتند روش های سنتی حمله می باشند. بر خلاف WEP که تکنیک های حمله مدرن مختلفی برای آن وجود داشته و بررسی شد، در مورد WPA اگر چه نا امنی آن اثبات شده و جایگزین کردن آن با WPA2 همواره توصیه شده است اما هنوز روش حمله کاملاً کاربردی یا اصطلاحاً Practical برای آن بصورت عمومی منتشر نشده است. تمامی تحقیقات انجام شده و حملات پیاده سازی شده تا تاریخ نگارش این مطلب در محیط و شرایط واقعی چندان کاربردی نبوده و استفاده از

آنها نیاز به فراهم بودن شرایط خاص دارد. اما با توجه به پیشرفت هایی که در تکنیک های حمله حاصل شده است، پیاده سازی حملات بصورت Practical و در قالب ابزارهایی با کاربری آسان مشابه آنچه در مورد WEP اتفاق افتاد، در آینده ایی نزدیک زیاد هم دور از انتظار نیست. در بخش های قبلی تفاوت های WPA با WEP مورد بررسی قرار گرفت و روشن شد که چرا تکنیک های قدیمی حملات مدرنن دیگر بر روی این پروتکل جدید تر قابل استفاده نمی باشند. این بخش به معرفی معدود تکنیک های حمله به WPA می کند که تاکنون بصورت عمومی منتشر شده اند.

Tews & Beck Attack:

در اواخر سال 2008 دو محقق رمزنگاری بنام های Martin Beck و Erik Tews اولین حمله علیه مکانیزم رمزنگاری (Cryptographic Attack) مورد استفاده در WPA یعنی TKIP را منتشر کردند. پیش از این تنها روش های شناخته شده عمومی برای حمله به WPA/2 مواردی بود که مورد بحث قرار گرفت. در مقاله[۶۹] معرف تکنیک جدید، روشی عملی برای بدست آوردن محتوای رمزشده یک پکت wpa مطرح شده است. برای درک این روش حمله لازم است یکبار دیگر تکنیک حمله Korek یا همان Chop-chop را که در بخش حملات پیشرفته علیه WEP مطرح شد مرور کنید. اساس این حمله پیاده سازی شده نیز مبتنی بر تکنیک مورد استفاده در Chop-Chop می باشد. اما تفاوت هایی مهم بین WEP و WPA وجود دارند. اگر بخاطر داشته باشید در WEP بدلیل عدم وجود هر نوع محدودیت در ارسال و تعداد خطاهای تولید شده، بسرعت می توان توسط روش KoreK مقدار PRGA را بدست آورد. در پروتکل WPA راهکاری برای مقابله با این حمله اضافه شد بدین صورت که با پیاده سازی مکانیزم حفاظتی Message Integrity Control (MIC) که بنام Michael نیز شناخته می شود، در صورت مشاهده دو خطای متوالی در Checksum (CRC32) در WEP را بیاد آورید) ارتباط بصورت خودکار قطع می شود. در مکانیزم کاری MIC در صورتی که کلاینت این خطاها را دریافت کرده باشد پس از 60 ثانیه توقف در ارتباط، بر اساس PMK اولیه، یک PTK جدید تولید می گردد. تولید PTK جدید به معنی شروع مجدد رمزنگاری توسط یک کلید جدید است. در صورتی که اکسس پوینت این خطاها را دریافت کرده باشد پس از 60 ثانیه توقف در ارتباط، عمل شروع رمزنگاری با کلید جدید را برای کلیه کلاینت های متصل شده انجام می دهد. با توجه به اینکه حمله KoreK مبتنی بر ارسال و ایجاد خطی برای استخراج PRGA است بنا بر این جلوی این حمله گرفته می شود. اما در روش حمله جدید به WPA راهکاری نیز برای عبور از این محدودیت پیدا شده که جزئیات آن بشرح زیر است.

تقریباً تمامی اکسس پوینت ها از قابلیت و استانداردی بنام 802.11e Quality of Service (QoS) پشتیبانی می کنند. در این استاندارد با ارسال داده های (پروتکل های) متفاوت بر روی کانال های قراردادی متفاوت، امکان طبقه بندی و اولویت بندی داده ها و ترافیک شبکه بوجود میاید. بطور مثال می توان مقرر کرد که پکت ها و داده های مربوط به VoIP از نظر در اختیار گرفتن پهنای باند و یا زمان پردازش و ارسال توسط روتر (اکسس پوینت) دارای ارجهیت بیشتری نسبت به سایر پروتکل ها باشند. در این استاندارد 8 کانال مختلف برای ارسال ترافیک مقرر شده است که به هر یک از این کانال ها Ttraffic Identifier (TID) گفته می شود. ضعف امنیتی که Martin Beck بر روی این پیاده سازی کشف کرد این است که مکانیزم امنیتی MIC که روش کار آن توضیح داده شد، تعداد خطاهای شناسایی شده را برای هر کانال بصورت مجزا شمارش و کنترل می کند! در استاندارد و پیاده سازی پروتکل WPA فرض بر این است که پس از مشاهده خطای دوم در Checksum پکت ها مکانیزم MIC فعال شده و عمل ReKeying صورت می پذیرد. اما با کشف این نکته مشخص شد که مکانیزم MIC درست پیاده سازی نشده است و می توان به ازای هر کانال QoS دو خطای CheckSum داشت یعنی در مجموع بجای 2 خطا، 16 خطا در دقیقه مجاز است. همچنین، محدودیتی که در پروتکل TKIP برای مقابله با باز-ارسال پکت های تکراری با Keystream یکسان پیاده سازی شده، با تکیه بر همین نکته مورد حمله قرار می گیرد. بدین ترتیب می توان پکت هایی جعلی و با Keystream یکسان را به تعداد کانال های QoS موجود تکرار کرده و ارسال کرد. با تکیه بر همین موارد، امکان انجام حمله Chop-Chop مجدداً میسر می شود البته نه با آن سرعتی که بر روی WEP قابل انجام است. لازم به یادآوری مجدد است که حمله Chop chop موجب شکسته شدن و استخراج کلید اصلی رمزنگاری

(PMK) نمی شود و توسط این حمله تنها محتویات رمزشده یک پکت خاص بازیابی می گردد و با بدست آمدن PRGA مورد استفاده برای رمزنگاری آن پکت، امکان جعل و ارسال مجدد آن فراهم می شود.

چندین نکته وجود دارد که باعث می شود روش حمله Tews & Beck هنوز آنطور که باید قابلیت استفاده در شرایط واقعی و Practical را نداشته باشد. اولین مورد این است که این حمله هنوز بصورت دو طرفه پیاده سازی نشده، یعنی با این روش تنها می توان ترافیک ارسالی از سمت اکسس پوینت به کلاینت را مورد حمله قرار داد و تهدیدی متوجه ترافیک ارسالی از کلاینت به اکسس پوینت نیست. علت اینکه این حمله تنها قابل انجام بر روی ترافیک اکسس پوینت به کلاینت است این می باشد که در صورت بروز خطا در MIC تنها اکسس پوینت است که پیغام خطایی در مورد این مشکل تولید و ارسال می کند. حمله کننده نیز بر اساس همین پیغام های خطا قادر به ادامه کار خود خواهد بود. در خصوص کلاینت ها، با توجه به اینکه پیغام خطایی در مورد MIC تولید و ارسال نمی گردد، حمله کننده عملاً راهی برای شناسایی شکست یا موفقیت مراحل کار خود در حمله ندارد و بنا بر این حمله به ترافیک ارسالی از کلاینت به اکسس پوینت بدین روش میسر نیست. مورد مهم دیگر QoS است که بصورت پیش فرض بر روی اکسس پوینت ها فعال نمی باشد. البته باید توجه داشت که اغلب اکسس پوینت هایی که از پروتکل 802.11n پشتیبانی و استفاده می کنند، QoS بعنوان جزئی از استاندارد 802.11n بر روی آنها فعال است، اما در سایر موارد کاربر اکسس پوینت می بایست در صورت نیاز QoS را تنظیم و فعال کند. در نهایت در صورتی که عمل ReKeying در خلال حمله به هر دلیل اتفاق بیافتد می بایست حمله مجدداً از ابتدا آغاز گردد. در اکسس پوینت ها در زمان استفاده از WPA/2 جدا از شرایطی که مکانیزم MIC فعال می گردد، خود اکسس پوینت عمل ReKeying را در بازه های زمانی مشخص انجام می دهد. این کار عموماً هر 3600 ثانیه یکبار انجام می شود اما مقدار فاصله زمانی قابل تغییر است. بنا بر این اگر فاصله زمانی بین Rekeying ها کوتاه تر از زمان لازم برای تکمیل حمله Chop-chop باشد، انجام این حمله عملاً غیر ممکن خواهد بود. 3600 ثانیه که مقدار پیش فرض تنظیم شده در اکسس پوینت ها می باشد بسیار بیشتر از زمان لازم برای انجام تکمیل حمله عنوان شده است.

با توجه به اینکه در حمله chop chop پکت بایت به بایت رمزگشایی می شود و همچنین محدودیت نیز در تعداد پکت های ارسالی با Checksum اشتباه وجود دارد، بنا بر این تنها پکت های کوچکی مانند درخواست های ARP و یا DNS مناسب برای حمله هستند. پس از رمزگشایی و بدست آمدن PRGA نفوذگر می تواند با جعل پکت، درخواست های دلخواه خود را به ترافیک تزریق کند. حتی با وجودی که این حمله در حال حاضر فقط پکت های ترافیک اکسس پوینت به کلاینت را رمزگشایی می کند، امکان پیاده سازی حملات ARP Spoofing/Poisoning و یا DNS Spoofing وجود دارد. در WPA و با وجود مکانیزم امنیتی که برای مقابله باز-ارسال پکت ها پیاده سازی شده، دیگر امکان پیاده سازی حملات ARP Request Replay وجود ندارد. بنا بر این استخراج کلید اصلی رمزنگاری مشخص شده توسط اکسس پوینت نیز تا به امروز وجود ندارد.

Tewsو Beck برای نشان دادن این روش حمله بصورت عملی، با اعمال تغییراتی در ابزار Aireplay-ng از بسته Aircrack، ابزار جدیدی را بنام TKIPtun-ngمنتشر کردند که یکی از روش های حمله معرفی شده در قالب تحقیق آنها در آن پیاده سازی شده است. ابزار TKIPtun-ng در واقع پیاده سازی حمله Chop-chop مطابق با شرایط و محدودیت های WPA می باشد، که قابلیت سو استفاده از QoS برای بهبود سرعت حمله در آن اضافه شده. اگر چه این دو محقق عنوان کردند که حتی در صورت عدم فعال بودن QoS نیز امکان پیاده سازی حمله آنها وجود دارد (این مورد در مقاله آنها بصورت کلی شرح داده شده است) اما تکنیک مذکور برای حمله بدون نیاز به فعال بودن QoS در نسخه عمومی ابزار آنها پیاده سازی نشده و در دسترس قرار ندارد.

TKIPtun-ng در حال حاضر در صورت موفقیت آمیز بودن حمله، قادر به رمزگشایی یک پکت ARP می باشد و پس از تکمیل حمله بمنظور اثبات موفقیت، پکت ARP رمزگشایی شده را مجدداً (و از طریق یکی از کانال های QoS) به ترافیک شبکه تزریق می کند. با استفاده از

PRGA بدست آمده نفوذگر می تواند یک پکت ARP جعلی رمز شده و معتبر تولید کرده و به ترافیک شبکه تزریق کند. تولید پکت با PRGA بدست آمده توسط ابزار packetforge که پیش از این مورد استفاده قرار گرفت قابل انجام است. پس از تهیه پکت (های) مورد نظر امکان پیاده سازی حملات ARP Poisoning فراهم می شود. بدلیل پیچیده و مشکل بودن نحوه انجام حمله بدین روش توسط ابزارهای موجود (بصورت نیمه خودکار) از ذکر جزئیات انجام حمله ARP Spoofing بر روی WPA در این متن صرف نظر میگردد. تا زمان نگارش این متن هنوز هیچ ابزاری که حمله مذکور را بصورت خودکار و پایدار بانجام برساند بصورت عمومی منتشر نشده است اما حتی با ابزارهای عمومی و در دسترس موجود نیز نفوذگری که از سطح دانش کافی برخوردار باشد قادر به پیاده سازی حملات ARP Poisoning بر روی شبکه های محافظت شده توسط WPA-TKIP خواهد بود.

در صورتی که قصد آزمایش حمله Tews & Beck را بصورت عملی در محیط آزمایشگاهی خود دارید، پس از فعال کردن QoS بر روی اکسس پوینت می توانید بروشی که در ادامه عنوان خواهد شد از ابزار TKIPtun-ng استفاده کنید. استاندارد QoS در تنظیمات برخی از اکسس پوینت ها بنام WiFi Multi-Media (WMM) معرفی شده و در دسترس است. این ابزار در نسخه پایدار بسته Aircrack-ng وجود ندارد و برای استفاده از آن می بایست آخرین نسخه بسته، از طریق SVN دریافت و کامپایل گردد. همچنین نسخه موجود در SVN در صورت استفاده از درایور MadWiFi بسیار ناپایدار عمل کرده و در بسیاری از موارد ممکن است پیش از اتمام پروسه حمله ابزار دوچار اختلال گردد. یکی دیگر از مشکلات نسخه در دسترس از طریق SVN مربوط به ضعف در پیاده سازی مکانیزم حمله می باشد. در حال حاضر بدلیل عدم تشخیص صحیح پیغام های خطای مربوط به فعال شدن راهکار امنیتی MIC، ممکن است ابزار در طول حمله پیغام خطا صادر شده از سوی اکسس پوینت را دریافت نکرده و در نتیجه به کار خود ادامه دهد. این امر سبب می شود تا ابزار از محدودیت تولید 2 خطا در دقیقه تجاوز کرده و در نتیجه موجب فعال شدن پروسه ReKeying توسط اکسس پوینت گردد. در چنین حالتی ابزار حمله خود را برای بدست آوردن PRGA ایی که دیگر معتبر نیست ادامه داده و در نتیجه حمله با شکست مواجه خواهد شد.

در این حمله بر خلاف حمله قبلی Chop-Chop، جعل کردن MAC Address و استفاده از MAC یک کلاینت که در حال حاضر به اکسس پوینت متصل است الزامی می باشد. بدین منظور پس از شنود ترافیک و بدست آوردن یک MAC معتبر، حمله کننده می بایست MAC خود را جعل و به این آدرس بدست آمده تغییر دهد.

در زیر مثالی از نحوه استفاده از ابزار و همچنین نمونه ایی از خروجی آن (برگرفته از وب سایت ابزار) آورده شده است.

```
Tkiptun-ng –h {Valid client MAC} –a {BSSID} –m 80 –n 100 ath0
```

```
#tkiptun-ng -h 00:0F:B5:AB:CB:9D -a 00:14:6C:7E:40:80 -m 80 -n 100 rausb0


The interface MAC (00:0E:2E:C5:81:D3) doesn't match the specified MAC (-h).
    ifconfig rausb0 hw ether 00:0F:B5:AB:CB:9D
Blub 2:38 E6 38 1C 24 15 1C CF
Blub 1:17 DD 0D 69 1D C3 1F EE
Blub 3:29 31 79 E7 E6 CF 8D 5E
15:06:48  Michael Test: Successful
15:06:48  Waiting for beacon frame (BSSID: 00:14:6C:7E:40:80) on channel 9
15:06:48  Found specified AP
15:06:48  Sending 4 directed DeAuth. STMAC: [00:0F:B5:AB:CB:9D] [ 0| 0 ACKs]
15:06:54  Sending 4 directed DeAuth. STMAC: [00:0F:B5:AB:CB:9D] [ 0| 0 ACKs]
15:06:56  WPA handshake: 00:14:6C:7E:40:80 captured
15:06:56  Waiting for an ARP packet coming from the Client...
Saving chosen packet in replay_src-0305-150705.cap
15:07:05  Waiting for an ARP response packet coming from the AP...
Saving chosen packet in replay_src-0305-150705.cap
15:07:05  Got the answer!
```

```
15:07:05  Waiting 10 seconds to let encrypted EAPOL frames pass without interfering.

15:07:25  Offset   99 ( 0% done) | xor = B3 | pt = D3 |  103 frames written in 84468ms
15:08:32  Offset   98 ( 1% done) | xor = AE | pt = 80 |   64 frames written in 52489ms
15:09:45  Offset   97 ( 3% done) | xor = DE | pt = C8 |  131 frames written in 107407ms
15:11:05  Offset   96 ( 5% done) | xor = 5A | pt = 7A |  191 frames written in 156619ms
15:12:07  Offset   95 ( 6% done) | xor = 27 | pt = 02 |   21 frames written in 17221ms
15:13:11  Offset   94 ( 8% done) | xor = D8 | pt = AB |   41 frames written in 33625ms
15:14:12  Offset   93 (10% done) | xor = 94 | pt = 62 |   13 frames written in 10666ms
15:15:24  Offset   92 (11% done) | xor = DF | pt = 68 |  112 frames written in 91829ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:18:13  Offset   91 (13% done) | xor = A1 | pt = E1 |  477 frames written in 391139ms
15:19:32  Offset   90 (15% done) | xor = 5F | pt = B2 |  186 frames written in 152520ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:22:09  Offset   89 (16% done) | xor = 9C | pt = 77 |  360 frames written in 295200ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:26:10  Offset   88 (18% done) | xor = 0D | pt = 3E |  598 frames written in 490361ms
15:27:33  Offset   87 (20% done) | xor = 8C | pt = 00 |  230 frames written in 188603ms
15:28:38  Offset   86 (21% done) | xor = 67 | pt = 00 |   47 frames written in 38537ms
15:29:53  Offset   85 (23% done) | xor = AD | pt = 00 |  146 frames written in 119720ms
15:31:16  Offset   84 (25% done) | xor = A3 | pt = 00 |  220 frames written in 180401ms
15:32:23  Offset   83 (26% done) | xor = 28 | pt = 00 |   75 frames written in 61499ms
15:33:38  Offset   82 (28% done) | xor = 7C | pt = 00 |  141 frames written in 115619ms
15:34:40  Offset   81 (30% done) | xor = 02 | pt = 00 |   19 frames written in 15584ms
15:35:57  Offset   80 (31% done) | xor = C9 | pt = 00 |  171 frames written in 140221ms
15:37:13  Offset   79 (33% done) | xor = 38 | pt = 00 |  148 frames written in 121364ms
15:38:21  Offset   78 (35% done) | xor = 71 | pt = 00 |   84 frames written in 68872ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:40:55  Offset   77 (36% done) | xor = 8E | pt = 00 |  328 frames written in 268974ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:43:31  Offset   76 (38% done) | xor = 38 | pt = 00 |  355 frames written in 291086ms
15:44:37  Offset   75 (40% done) | xor = 79 | pt = 00 |   61 frames written in 50021ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:47:05  Offset   74 (41% done) | xor = 59 | pt = 00 |  269 frames written in 220581ms
15:48:30  Offset   73 (43% done) | xor = 14 | pt = 00 |  249 frames written in 204178ms
15:49:49  Offset   72 (45% done) | xor = 9A | pt = 00 |  183 frames written in 150059ms
Looks like mic failure report was not detected. Waiting 60 seconds before trying again to avoid the AP shutting down.
15:52:32  Offset   71 (46% done) | xor = 03 | pt = 00 |  420 frames written in 344400ms
15:53:57  Offset   70 (48% done) | xor = 0E | pt = 00 |  239 frames written in 195980ms
Sleeping for 60 seconds.36 bytes still unknown
ARP Reply
Checking 192.168.x.y
15:54:11  Reversed MIC Key (FromDS): C3:95:10:04:8F:8D:6C:66

Saving plaintext in replay_dec-0305-155411.cap
Saving keystream in replay_dec-0305-155411.xor
15:54:11
Completed in 2816s (0.02 bytes/s)

15:54:11  AP MAC: 00:40:F4:77:F0:9B IP: 192.168.21.42
15:54:11  Client MAC: 00:0F:B5:AB:CB:9D IP: 192.168.21.112
15:54:11  Sent encrypted tkip ARP request to the client.
15:54:11  Wait for the mic countermeasure timeout of 60 seconds.
```

نکته مهم در این حمله محدود کردن حداقل و حداکثر سایز پکت هایی است که ابزار آنها را برای رمزگشایی و حمله انتخاب می کند. پکت های بزرگ باعث طولانی شدن تعداد بررسی های صحیح و خطا و همچنین طولانی شدن زمان حمله می شوند که هر دو شانس موفقیت حمله را کاهش می دهند. علت حمله به پکت های ARP نیز همین است. در پکت های ARP رمز شده تعداد بایت های نامعلوم پکت بسیار کم است و در نتیجه مکانیزم حمله Chop-Chop نیاز به تعداد تلاش های کمتری برای کشف مقادیر این بایت ها دارد. تکمیل حمله و رمزگشایی برای بدست آوردن PRGA توسط این ابزار به حدود 15 دقیقه زمان نیاز دارد.

بهبود حمله Tews & Beck (1): در سال 2009 نتیجه تحقیقی منتشر شد که طی آن[۷۰]، روش هایی برای بهبود تکنیک های حمله به TKIP و همچنین حمله Tews & Beck معرفی و مورد بحث قرار گرفت. تحقیق انجام شده از دو نظر دارای اهمیت می باشد. مورد اول اینکه بر اساس نتایج بررسی شده در قالب این تحقیق مشخص می شود که بجز پکت های ARP (بدلیل مشخص بودن تعداد زیادی از بایت های هر پکت) می توان حمله را بر روی پکت های پروتکل های دیگری مانند DHCP و یا DNS نیز بصورت موثر انجام داد. بدین ترتیب دامنه حملات ممکن توسط این روش (بصورت Practical) گسترش می یابد. این به معنی بدست آوردن تعداد بایت های بیشتری از Keystream مورد استفاده در رمزنگاری نیز می باشد، بطوری که در حمله اولیه تنها 48 بایت از Keystream بدست می آمد اما در این روش بهبود یافته 596 بایت از Keystream بدست می آید. همچنین حمله ARP Spoofing پیاده سازی شده در ابزار tkiptun-ng تنها جنبه اثبات (Proof of Concept) داشته و در عمل تهدیدی را ایجاد نمی کند. در روال این تحقیق انجام یک حمله ARP Poisoning بصورت عملی مورد بررسی قرار گرفته و بانجام رسیده است.

مورد دوم قابل توجه، رفع برخی از اشکالات مهم حمله اولیه Tews & Beck در پیاده سازی حمله در قالب ابزار tkiptun-ng می باشد. همانطور که بر بخش قبلی گفته شد یکی از اشکالات مهم این ابزار ضعف در تشخیص و مدیریت پیغام های خطای (MIC Failur) تولید شده توسط اکسس پوینت بود که موجب شکست حمله می گردد. با اعمال تغییراتی در کد ابزار، این مشکل (و برخی مشکلات جزئی دیگر) رفع شده و پایداری و شانس موفقیت حمله پس از این تغییرات تا حد زیادی افزایش یافته است. لازم به ذکر است که بسیاری از این تغییرات در نسخه جاری SVN ابزار لحاظ شده است، اما تکنیک های حمله جدید (مانند حمله به DHCP و جعل پکت های آن) به کد اصلی و رسمی ابزار اضافه نشده است.

در نهایت، علاوه بر روش ها و بهبود های ذکر شده، این تحقیق به معرفی روشی برای پیاده سازی حمله DoS علیه اکسس پوینت می نماید. در حمله معرفی شده با سو استفاده از مکانیزم امنیتی MIC حمله کننده قادر خواهد بود تا کارایی کل شبکه (کلاینت های متصل به اکسس پوینت) را تحت تأثیر قرار داده و در برقراری ارتباط اختلال ایجاد کند. بحث پیاده سازی حملات DoS بر روی شبکه های بیسیم بسیار گسترده بوده و خود نیازمند توضیحی جدا و مفصل می باشد. نکته قابل توجه در مورد این روش حمله DoS سو استفاده آن از MIC و حمله به مکانیزم رمزنگاری TKIPمی باشد. در شرایط معمول می توان با تزریق پکت های جعلی De-Authentication کلاینت های شبکه را وادار به قطع ارتباط نمود و این کار با ارسال مداوم پکت ها از سوی حمله کننده صورت می پذیرد. در این روش جدید، با تولید مکرر خطا و فعال کردن مکانیزم MIC حمله کننده سبب می شود تا اکسس پوینت دائماً پروسه امنیتی قطع ارتباط بمدت 60 ثانیه را تکرار کند. در چنین شرایطی اکسس پوینت پس از هر بار فعال شدن MIC، تنها 22 ثانیه (بر طبق محاسبه عنوان شده در متن منتشر شده) فعال بوده و سرویس دهی می کند و این 22 ثانیه زمانی است که حمله کننده برای تکرار دور جدید DoS و فعال سازی دوباره MIC نیاز دارد. جزئیات این حمله و کد های لازم برای Patch کردن ابزار tkiptun-ng در متن منتشر شده ارائه گردیده است. علاوه بر خود مقاله، کد های لازم برای پیاده سازی حملات مورد بحث از طریق وب سایت Track[۷۱] بسته ابزار Aircrack-ng نیز در دسترس قرار دارد.

بهبود حمله Tews & Beck (2) : در اواسط سال 2009 و پس از تحقیق ذکر شده در بخش قبل، دو محقق ژاپنی در قالب یک کنفرانس و مقاله[۷۲] ادعا کردند که تکنیک حمله جدیدی را کشف کرده اند که زمان لازم برای حمله به WPA را کاهش داده و عنوان شد که با بهبود تکنیک حمله Tews & Beck میزان زمان لازم برای حمله به به WPA را می توان تا یک دقیقه کاهش داد ! این ادعا اگرچه به نوعی غیر واقع بوده و عملاً اشتباهی (شاید عمدی) در روش تبلیغ برای معرفی این تکنیک جدید توسط رسانه ها بوده است، در هر حال راهکاری جدید برای حمله به TKIP معرفی شد اما زمان لازم برای پیاده سازی آن عملاً بسیار بیشتر از زمان تعجب برانگیز تبلیغ شده توسط رسانه هاست.

روش حمله جدید، مبتنی بر استفاده از MITM (Man In The Middle) Attack می باشد و در سناریو عنوان شده، حمله کننده بعنوان اکسس پوینت، واسط ارتباط کلاینت با اکسس پوینت هدف خواهد بود. همچنین این تکنیک حمله، بر خلاف تکنیک اولیه مطرح شده توسط Tews & Beck قادر به حمله به اکسس پوینت های بدون QoS نیز می باشد. بحث های زیادی در مورد این تکنیک حمله مطرح شده است از جمله آن می توان به مشکل بودن پیاده سازی سناریو حمله و سختی پیاده سازی آن در شرایط واقعی اشاره نمود. اگر بخواهیم مکانیزم کاری این روش حمله را به زبانی بسیار ساده عنوان کنیم این حمله تکنیک Chop-Chop مطرح شده توسط Tews & Beck را بواسطه استفاده از MITM پیاده می کند و در چنین حالتی پیغام های MIC Failur (که موجب فعال شدن مکانیزم MIC و قطع ارتباط و ReKeying می شوند) توسط حمله کننده به کلاینت انتقال داده نمی شوند. در نتیجه می توان در زمان کوتاه تری حمله Chop-Chop را بانجام رساند. بر طبق آنچه در مقاله عنوان شده، این حمله در سه مرحله اتفاق می افتد.

در مرحله اول، حمله کننده می بایست کلاینت را وادار به استفاده از وی بعنوان واسط و اکسس پوینت نماید. این کار اصطلاحآ بعنوان روش پیاده سازی Rouge Access Point نیز شناخته می شود. در چنین حالتی که مقاله تحت عنوان "Repeater Mode" از آن یاد کرده حمله کننده کلیه پکت های ارسالی مربوط به SSID تحت حمله را بدون تغییر انتقال می دهد.

در مرحله دوم که تحت عنوان "MIC Key Recovery Mode" ذکر شده، با استفاده از مکانیزم حمله Chop-Chop حمله کننده Keystream را بدست می آورد. این مرحله همانند روش اصلی حمله به 12 تا 15 دقیق زمان نیاز دارد.

در نهایت و در مرحله سوم که بنام "Message Flasification Mode" معرفی شده حمله اصلی و تکنیک جدید معرفی شده بکار برده می شود تا با استفاده از آن پکت های ARP جعلی تولید و به ترافیک تزریق گردند. در روش های مطرح شده قبلی این کار (تولید یک بسته ARP جعلی با استفاده از Keystream بدست آمده) به حدود 4 دقیقه زمان نیاز دارد زیرا 4 بایت از پکت ARP مجهول بوده و مقدار آنها می بایست بروش Chop-Chop محاسبه گردد. محاسبه هر بایت نیز با توجه به محدودیت اعمال شده توسط MIC در خصوص تعداد خطاهای مجاز، به یک دقیقه زمان احتیاج دارد. در این تکنیک جدید آنطور که عنوان شده، با توجه به اینکه آدرس IP اکسس پوینت در مرحله قبل (MIC Key Recovery Mode) بدست آمده است، میزان بایت های مجهول پکت های ARP رمز شده به یک بایت کاهش میابد. در این حالت می توان بدون محدودیت و مواجه شدن با مکانیزم امنیتی MIC اقدام به حدث زدن آن یک بایت توسط روش Chop-Chop نمود. این کار به یک دقیقه زمان نیاز دارد. این مدت زمان کوتاه که بسیاری از آن برای معرفی این تکنیک حمله استفاده کردند در واقع زمان لازم برای پیاده سازی آخرین مرحله حمله می باشد !

لازم به ذکر است که جزئیات زیادی در مورد این روش جدید بصورت عمومی منتشر نشده و محققین آن نیز تا زمان نگارش این مطلب هیچ مثال عملی و یا ابزاری برای اثبات تکنیک خود منتشر نکرده اند. بنا بر این تا به امروز این تکنیک حمله تنها بصورت تئوری مطرح شده است. در برخی از فروم ها و یا لیست های پستی بحث هایی پیرامون این تکنیک و (نا)کارامدی آن مطرح شده که مطالعه آنها به درک بیشتر روش حمله کمک می کند. یک نمونه از این دست، عنوان "WPA Attack Improved to 1 min"[۷۳] در لیست پستی DailyDave می باشد.

سخن پایانی : در این مقاله، بسیاری از روش های رایج که بصورت عمومی منتشر شده و برای حمله به شبکه های بیسیم از طریق ضعف های امنیتی پروتکل های WEP/WPA/WPA2 مورد استفاده قرار می گیرند مورد بحث قرار گرفته و معرفی شدند. اگر چه معرفی حملات بصورت سطحی بوده و تنها بمنظور آشنایی خواننده با تکنیک های موجود بوده است اما آگاهی از جزئیات کامل هر تکنیک حمله و تسلط کامل بر آن مستلزم مطالعه جزئیات منتشر شده برای هر مورد می باشد. هدف این مقاله نیز آشنا کردن ذهن خواننده با این موارد و معرفی

منابعی برای مطالعه کامل در خصوص هر روش حمله بوده است. همچنین موارد و تکنیک های عنوان شده در این مقاله تنها بر روی حمله مستقیم به بستر شبکه از طریق اکسس پوینت و پروتکل های امنیتی مربوطه اشاره دارد. تکنیک های حمله دیگری نیز وجود داشته و مرسوم هستند اما تمرکز آنها بر روی حمله به کلاینت های شبکه و یا بستر نرم افزاری (بعنوان مثال Device Driver های تجهیزات بیسیم) می باشد. با توجه به مشکل تر شدن حمله مستقیم به شبکه هایی که از WPA/2 استفاده می کنند و غیر عملی بودن برخی از تکنیک های معرفی شده در شرایط واقعی به دلایلی از جمله استفاده از کلمات عبور امن و همچنین ناپایدار بودن روش های حمله جدید، بسیاری از نفوذگران ترجیح می دهند تا در مواقع رویارویی با شبکه های بیسیم، از تکنیک های حمله Client-Side برای حصول دسترسی به شبکه استفاده کنند. امید است در صورت فراهم شدن مجالی دوباره، این تکنیک های حمله Client-Side نیز به همین صورت که در این مقاله برای WEP/WPA آورده شد، در مقاله ایی دیگر مورد بحث و بررسی قرار گیرند.

[1] http://www.slitaz.org/en/
[2] http://www.wirelessdefence.org/Contents/WirelessBuildHowtoFC6.htm
[3] http://www.cacetech.com/products/airpcap.html
[4] http://www.wildpackets.com/support/omni/omnipeek_basic/wireless
[5] http://www.wildpackets.com/products/network_analysis/omnipeek_network_analyzer/wireless_network_analysis
[6] http://www.tamos.com/products/commwifi/
[7] http://www.tamos.com/products/commwifi/adapterlist.php
[8] http://www.tamos.com/files/cardcheck.zip
[9] http://msdn.microsoft.com/en-us/library/aa503132.aspx
[10] http://www.inguardians.com/pubs/Vista_Wireless_Power_Tools-Wright.pdf
[11] http://en.wikipedia.org/wiki/EAPOL
[12] http://wiki.wireshark.org/HowToDecrypt802.11
[13] http://www.aircrack-ng.org/doku.php?id=airdecap-ng
[14] http://www.netstumbler.com/
[15] http://www.metageek.net/products/inssider
[16] http://www.vistumbler.net/
[17] http://iase.disa.mil/tools/index.html
[18] http://download.eeye.com/html/products/retinawireless/
[19] http://robota.net/
[20] http://www.kismetwireless.net/
[21] http://www.aircrack-ng.org/doku.php?id=airodump-ng
[22] http://www.remote-exploit.org/codes_wellenreiter.html
[23] http://community.corest.com/~hochoa/wifizoo/index.html
[24] http://madwifi-project.org/
[25] http://research.microsoft.com/en-us/downloads/994abd5f-53d1-4dba-a9d8-8ba1dcccead7/
[26] http://backtrack.offensive-security.com/index.php/Howto
[27] http://forums.remote-exploit.org/bt4beta-howtos/
[28] http://www.kismetwireless.net/documentation.shtml
[29] http://qkismet.sourceforge.net/
[30] http://www.aircrack-ng.org/doku.php#aircrack-ng_suite
[31] http://www.aircrack-ng.org/doku.php?id=airodump-ng
[32] http://www.aircrack-ng.org/doku.php?id=injection_test
[33] http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy
[34] http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf
[35] http://en.wikipedia.org/wiki/Rc4

[36] http://en.wikipedia.org/wiki/CRC-32
[37] http://en.wikipedia.org/wiki/Stream_cipher
[38] http://cnscenter.future.co.kr/resource/hot-topic/wlan/wep_attack.pdf
[39] http://wepcrack.sourceforge.net/
[40] http://www.cs.jhu.edu/~astubble/
[41] http://airsnort.shmoo.com/
[42] http://web.archive.org/web/20061026140735/www.dachb0den.com/projects/bsd-airtools/wepexp.txt
[43] http://weplab.sourceforge.net/
[44] http://wepattack.sourceforge.net/
[45] http://www.802.11mercenary.net/jc-wepcrack/
[46] http://www.802.11mercenary.net/jc-aircrack/
[47] http://www.netstumbler.org/f50/chopchop-experimental-wep-attacks-12489/
[48] http://en.wikipedia.org/wiki/ICV_-_Integrity_Check_Value
[49] http://www.informit.com/guides/printerfriendly.aspx?g=security&seqNum=197
[50] http://www.aircrack-ng.org/doku.php?id=fake_authentication
[51] http://darkircop.org/bittau-wep.pdf
[52] http://www.aircrack-ng.org/doku.php?id=fragmentation
[53] http://www.aircrack-ng.org/doku.php?id=arp-request_reinjection
[54] http://cage.ugent.be/~klein/RC4/
[55] http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/
[56] http://eprint.iacr.org/2007/120.pdf
[57] http://www.aircrack-ng.org/doku.php?id=aircrack-ng
[58] http://en.wikipedia.org/wiki/Message_Integrity_Code
[59] http://en.wikipedia.org/wiki/Temporal_Key_Integrity_Protocol
[60] http://en.wikipedia.org/wiki/802.11i
[61] http://en.wikipedia.org/wiki/802.1x
[62] http://trac.aircrack-ng.org/browser/branch/aircrack-ng-cuda
[63] http://www.elcomsoft.com/ewsa.html
[64] http://en.wikipedia.org/wiki/Rainbow_table
[65] http://www.wigle.net/gps/gps/main/ssidstats
[66] http://code.google.com/p/pyrit
[67] http://www.offensive-security.com/wpa-tables/
[68] http://www.renderlab.net/projects/WPA-tables/
[69] http://dl.aircrack-ng.org/breakingwepandwpa.pdf
[70] http://download.aircrack-ng.org/wiki-files/doc/tkip_master.pdf
[71] http://trac.aircrack-ng.org/ticket/684
[72] http://jwis2009.nsysu.edu.tw/location/paper/A%20Practical%20Message%20Falsification%20Attack%20on%20WPA.pdf
[73] http://lists.immunitysec.com/pipermail/dailydave/2009-August/thread.html