Journal Of Electronical & Cyber Defence
Vol. 2, No. 4, 2015, Serial No. 8

Utilizing Port-Knocking as first defensive layer at defense-in-depth strategies using hybrid of the Internet Control Message Protocol features, Internet Addresses and Tunneling

M. Pourvahab[1], R. Ebrahmi Atani[2]*

1- MSc. in Information Technology - Computer Networks, Pardis 2, University of Guilan, rasht, Iran.
2- Assistant professor in Computer Engineering Department of University of Guilan, Rasht, Iran.
(Received: 18/12/2013 , Accepted: 11/05/2015)

ABSTRACT

The computer networks are always vulnerable to various attacks and these attacks are typically include identification attacks, acquire attacks and disabling services attacks. At identification attacks, the attackers attempt to gather information and identify running services, in order to achieve damage, acquiring or disabling services. Port-Knocking (PKn) is a unique method to prevent detection and exploiting vulnerable services by the attackers and in facts the aim of PKn is hiding the services from attacker's view and combat identifying attacks, while the authenticated users are allowed to access these hidden services. In this article, a new method to establish simplicity and use of existing tools at the most operating systems to eliminate specific programs for running processes and open ports PKn at any time and anywhere have been introduced. This novel PKn can create more complexity at Knock operation utilizing the specific ICMP and synchronizing by the use of web browsers, to reduce of replay attacks and eliminate the risk of DoS attacks by hidden the services. To insure the efficiency and capabilities of the proposed method, this technique is successfully implemented and ran on a MikroTik RouterOS operation system.

Keywords: Network Security; Service Security; Port Security; Authentication; Port-Knocking.

* Corresponding Author Email: rebrahimi@guilan.ac.ir

مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال دوم، شماره ۴، زمستان ۱۳۹۳، ص ۴۰-۲۵

بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام های اینترنتی، آدرس اینترنتی و تونل‌زنی

مهران پوروهاب[۱]، رضا ابراهیمی آتانی[۲*]

۱- کارشناس ارشد مهندسی فناوری اطلاعات- شبکه‌های کامپیوتری پردیس دانشگاه گیلان، رشت، ایران

۲-استادیار، گروه مهندسی کامپیوتر، دانشگاه گیلان، رشت، ایران

(دریافت: ۹۲/۹/۲۷؛ پذیرش: ۹۴/۲/۲۱)

چکیده

شبکه‌های کامپیوتری همواره مستعد مبتلا شدن به انواع حملات بوده و این حملات به‌طور معمول شامل حملات شناسایی، دست‌یابی و از کار انداختن خدمات هستند. در حملات از نوع شناسایی، مهاجمین اقدام به جمع‌آوری اطلاعات و شناسایی خدمات در حال اجرا در جهت نیل به آسیب‌رسانی، دست‌یابی و یا از کار انداختن خدمات هستند. در واقع، خوراک حملات دست‌یابی و از کار انداختن خدمات از طریق حملات شناسایی فراهم می‌شود. Port-Knocking(PKn) یک روش منحصر بفرد برای جلوگیری از کشف و بهره‌برداری از خدمات آسیب پذیر توسط مهاجمین بوده و در واقع هدف آن مخفی نگه‌داشتن خدمات از دید نفوذگران و عدم کارآیی حملات شناسایی است، درحالی‌که کاربران تصدیق‌شده اجازه دسترسی به این خدمات نامرئی را دارند. در این مقاله روش جدیدی با هدف ایجاد سادگی و استفاده از ابزارهای موجود در اغلب سیستم‌عامل‌ها به‌منظور حذف برنامه‌های خاص و پیچیده جهت اجرای فرآیند PKn و باز کردن پورت‌ها در هر زمان و هر کجا معرفی گردیده است. از اهداف دیگر این روش، ایجاد پیچیدگی بیشتر در عملیات Knock با بکارگیری از ویژگی‌های خاص پروتکل ICMP و استفاده همزمان از مرورگرهای اینترنتی جهت کاهش کارآیی حملات Replay و از بین بردن مخاطرات ناشی از حملات DoS با مخفی نگه داشتن خدمات است. جهت اطمینان از کارآیی و قابلیت‌های ارائه‌شده، این روش با موفقیت بر روی سیستم عامل RouterOS میکروتیک پیاده‌سازی و اجرا گردیده است.

واژه‌های کلیدی: امنیت شبکه، امنیت خدمات، امنیت پورت، احراز هویت، پورت- ناکینگ.

۱- مقدمه

یکی از مشکلات مهم در حوزه امنیت شبکه، مجموعه سرویس‌های در حال اجرا بر روی دستگاه‌های شبکه، علی‌الخصوص پورت‌های باز است،که به همه اشخاص اجازه می‌دهد تا به آن سرویس‌ها متصل شوند و این امکان را نیز فراهم می‌سازند تا از طریق یکی از راه‌های بی‌شمار نفوذ، به این سرویس‌ها حمله شود. اینترنت در دهه ۱۹۸۰ و در سال‌های اولیه کار خود صرفاً با ذهنیت ایجاد قابلیت همکاری شکل گرفت و مهندسان می‌خواستند ماشین‌هایی را برای برقراری ارتباط آسان از طریق برخی از محدودیت‌ها توسط ویژگی‌های امنیتی غیرضرروی از قبیل احراز هویت و کنترل دسترسی به ارمغان آورند. اما ویژگی‌های امنیتی نیز خالی از ایراد نبودند، در واقع نقص‌هایی که در برنامه‌نویسی نرم‌افزارهای فعال در شبکه به همراه نقاط ضعف دیگری که داشتند خیلی سریع فرصت‌های بسیاری را به مهاجمین برای حمله به شبکه‌ها دادند.

بدون تردید رشد سریع دنیای ارتباطات خصوصاً اینترنت، از دهه ۹۰ میلادی به بعد موجب آن گردیده که شبکه‌های کامپیوتری دارای ساختار بسیار پیچیده‌تر از گذشته شده و همچنین نرم‌افزارها بزرگتر، دارای ساختاری مرکب و در بسیاری از موارد نیز نیاز به اتصال به اینترنت داشته باشند. همان‌طور که در دنیای اینترنت، شبکه‌های ارتباطی و نرم‌افزارهای مورد استفاده در آن رشد نموده و

* رایانامه نویسنده مسئول: rebrahimi@guilan.ac.ir

پیشرفت داشته‌اند، خراب‌کاری‌ها و دستبردهای اینترنتی نیز روز به روز در حال پیشرفت و رو به افزایش است.

حال که حملات اینترنتی هر روزه در حال افزایش بوده و این موضوع نیز بر کسی پوشیده نیست که اینترنت یک محیط ناامن و خطرناک است و تأمین امنیت برای بهره برداری از این شبکه جزء ملزومات اصلی استفاده از آن شده است، لذا جایگاه و نقش امنیت در شبکه‌ها و یا هر کامپیوتر متصل به اینترنت پر رنگ‌تر از قبل شده است.

بنابراین در طول این سال‌ها بسیاری از پروتکل‌ها و حتی ساختارهای شبکه، بهبود و ارتقاء یافتند و هدف آن‌ها ایجاد سطوح مقاومتی قوی‌تر در مقابل حملات بوده است. یکی از دغدغه‌های اصلی مفهوم احراز هویت که راه‌حل‌های بسیاری نیز برای آن ارائه شده است، ملزم نمودن کاربر به وارد کردن نام کاربری و رمز عبور قبل از استفاده از دریافت خدمات از سرویس است که یکی از محبوب‌ترین و پرکاربردترین آن‌ها نیز محسوب می‌شود.

سرویس‌هایی نظیر Web یا Email نیاز دارند دائما در حال کار و ارائه خدمات به عموم باشند تا همگان بتوانند این سرویس‌ها را مشاهده و استفاده نمایند، اما گاهی برخی از سرویس‌ها یا پورت‌ها فقط در زمان خاصی مورد نیاز بوده و احتیاجی به خدمات دائمی آنها نیست [۱] و پس از خاتمه کار باید بسته گردند زیرا باز بودن و ارائه خدمات دائمی این سرویس‌ها، خطرساز بوده و ریسک امنیتی بالایی را ایجاد می‌نماید، یا اینکه گاهی مجبور به استفاده از برنامه‌ها و سرویس‌های قدیمی در اینترنت هستیم که قابلیت به‌روزرسانی نداشته و حتی دارای ساده‌ترین سیستم‌های امنیتی و احراز هویت نیز نیستند و یا اینکه برخی از سرویس‌ها ذاتا ناامن هستند، بدین معنی که ناامن طراحی و پیاده‌سازی شده‌اند و هیچ سازوکار احرازهویتی برای آنها طراحی نشده است [۲]. همچنین در مواقعی از یک سرویس جدید با رعایت اصول امنیتی در زمان ارائه محصول در حال استفاده هستیم، اما به‌طور دائم نمی‌توانیم چشم خود را به آسیب‌پذیری‌ها و حفره‌های امنیتی احتمالی که قرار است در آینده پیدا شود بدوزیم و درصورت ارائه اصلاحیه سریعاً آن را نصب کنیم [۱].

حفاظت از محرمانگی، یکپارچگی و احرازهویت به یک نگرانی اصلی برای پروتکل‌های حامل ترافیک‌های حساس تبدیل شده است. (امنیت اطلاعات معمولاً متکی بر سه مفهوم محرمانگی، یکپارچگی و در دسترس بودن است، با این حال در دسترس بودن به‌طور رسمی در نقل و انتقالات شبکه‌ها استفاده نمی‌شود.) بدون این اقدامات احتیاطی، اطلاعاتی که بر روی شبکه ارسال می‌شود ممکن است دستخوش تغییرات و یا در معرض خطر افشا شدن قرار گیرند.

به‌طور معمول، ابتدایی‌ترین خطوط دفاعی ما در مقابل حملات اینترنتی، استفاده از دیواره‌های آتش (Firewall) و به‌روز نگاه‌داشتن و استفاده از آخرین نسخه برنامه‌ها (با توجه به گفته بالا اگر برنامه قدیمی نبوده و قابلیت به‌روزرسانی داشته باشد) است. همچنین استفاده هم‌زمان از سیستم‌های تشخیص نفوذ/ جلوگیری از نفوذ (IDS/IPS) نیز می‌تواند بسیار مفید باشند که در نهایت منجر به ارتقاء سطح امنیت می‌شنوند [۳]. در بسیاری از موارد، مهاجمین و نفوذگران با ابزارهایی شبیه پورت اسکنرها در ابتدا به جمع‌آوری اطلاعات از شبکه هدف پرداخته و اطلاعات مواردی نظیر پورت‌های باز، سرویس‌های در حال اجرا و نرم‌افزارهای در حال استفاده را استخراج می‌نمایند. سپس نسبت به پیدا نمودن آسیب پذیری در نرم‌افزار و یا سرویس‌های در حال اجرا اقدام نموده و آن‌گاه به دور زدن سیستم از روش‌هایی مانند سرریز بافر و یا استفاده از آسیب‌پذیری‌ها مانند حملاتی شبیه zero-day مبادرت می‌کنند و ممکن است نقص و حفره موجود در سیستم موجب نقض یکی از موارد محرمانگی، یکپارچگی و یا در دسترس بودن گردد [۴]. همچنین مهاجمین می‌توانند با تغییر مبدأ آی‌پی خود، (Source IP Address) منشاء بسته را عوض نموده و با دست‌کاری بسته‌ها، محتوای دلخواه و مخرب خود را در داخل آن پنهان کرده و با ارسال آن دیواره‌های آتش را دور بزنند [۵].

بنابراین باز کردن پورت توسط هریک از سرویس‌ها و باز گذاشتن دائمی یک پورت، به عنوان یک تهدید در نظر گرفته می‌شود. درنتیجه، نظارت و کنترل دسترسی به پورت‌ها می‌تواند تضمینی قابل اعتماد جهت داشتن اتصال و ارتباطی امن باشد و به دنبال آن در ابتدایی‌ترین سطح در لبه شبکه علاوه بر سیستم‌هایی نظیر دیوارهٔ آتش، به مکانیزم‌های خوب و قوی‌تر دیگری نیز نیاز داریم. یکی از این مکانیزم‌ها احرازهویت است. راه‌ها و روش‌های زیادی جهت ایجاد سرویس‌های احراز هویت به‌منظور افزایش لایه‌های امنیتی وجود دارند. یکی از این روش‌ها Port-Knocking است که می‌تواند سرویس‌ها را از طریق انتقال اطلاعات برروی پورت‌های بسته، از دید مهاجمین مخفی نگه دارد.

تعاریف متعددی از Port-Knocking وجود دارد. B. Maddock این‌طور تعریف کرده است که " Port-Knocking یک روش انتقال اطلاعات از طریق پورت‌های

بسته بر روی یک کامپیوتر شبکه‌ای است [۶].“ و همچنین معروف ترین تعریف از M.Krzywinski است که می‌گوید ” Port-Knocking یک روش ایجاد اتصال به یک کامپیوتر شبکه‌ای است که هیچ پورت بازی ندارد [۷].“ و تعریف دیگری از M.Krzywinski می‌گوید ”Port-Knocking یک روش پنهانی انتقال اطلاعات در شبکه‌های کامپیوتری است [۸].“ از این پس به‌منظور سهولت بیشتر، از کلمه اختصاری PKn به‌جای Port-Knocking استفاده می‌کنیم.

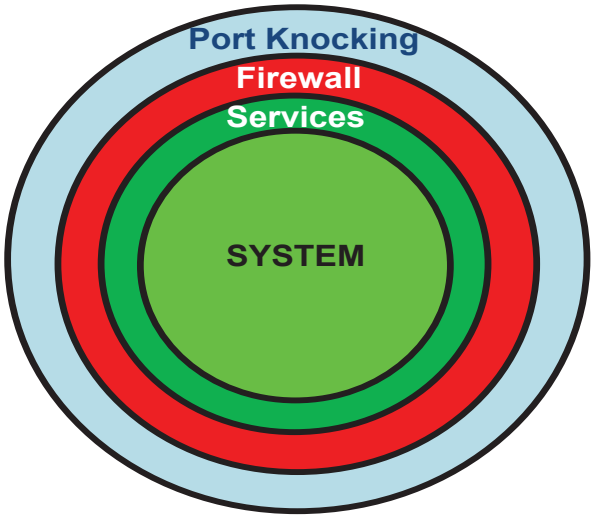
بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام‌های اینترنتی، آدرس اینترنتی و تونل‌زنی به‌عنوان یک روش بسیار کارآمد و قدرتمند در مقابل حملات روز- صفر (Zero -Day)، کاوشگرهای پورت (Port Scanners)، سرویس‌های خطرناک، سرویس‌هایی که هیچ‌گونه سازوکار امنیتی (نظیر احراز هویت) ندارند، آسیب‌پذیری‌های اصلاح نشده، حملات جستجوی فراگیر (brute force)، استراق سمع (sniffing)، و مشکلات استفاده از IP اشتراکی در شبکه‌های پشت سرویس NAT می‌باشد، که به‌عنوان خط مقدم دفاع در مقابل حملات اینترنتی ایفای نقش می‌کند. علی‌رغم قابلیت‌های فراوان ذکرشده در بالا، این روش یک سازوکار بسیار سبک‌وزن است. درنتیجه دارای کمترین سربار بر روی سیستم سرویس‌دهنده بوده به‌طوری‌که از حداقل منابع موجود در سیستم استفاده می‌کند. از نقاط متمایز کننده این روش با روش‌های دیگر، باز کردن پورت یا سرویس در هر زمان و هر مکان با استفاده از ابزارهای در دسترس نظیر مرورگرهای اینترنتی و ابزاری نظیر Ping است.

در ادامه مطالب، مقاله بر اساس این ساختار ارائه خواهد شد: در بخش۲، نقاط قوت و ضعف PKn بررسی شده و PKn پایه نیز به‌صورت کامل توصیف و کامل‌ترین تاریخچه از PKn که تا کنون منتشر نشده است ارائه می‌شود. در بخش ۳ ، ضمن مطالعه و بررسی مقالات قبلی این حوزه، ایده‌ها و روش‌های پیاده‌سازی‌شده قبلی از حیث نقاط قوت و ضعف مورد بررسی قرار می‌گیرند. در بخش ۴ به بعد نسبت به تشریح روش جدیدی از PKn، بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام‌های اینترنتی (ICMP)، آدرس اینترنتی (URL) و تونل‌زنی (Tunneling) اقدام می‌شود. در بخش ۵ نتایج پیاده‌سازی این طرح ارائه خواهد شد و مقاله در بخش ۶ جمع‌بندی می‌گردد.

۲- Port Knocking

در شبکه‌های کامپیوتری PKn به یک روش بازکردن پورت از بیرون شبکه بر روی دیوارهٔ آتش از طریق ایجاد یک‌سری اتصال اطلاق می‌شود که می‌تواند با نظم و ترتیب خاصی بر روی مجموعه‌ای از پورت‌های بسته ازپیش تعریف‌شده اعمال گردد [۹]. به‌عبارت دیگر، PKn یک روشی از احراز هویت است که به‌منظور احراز هویت از پورت‌های بسته جهت انتقال اطلاعات استفاده می‌کند؛ در واقع PKn اطلاعات را بر روی پورت‌های بسته ارسال می‌نماید. آنچه اتفاق می‌افتد این است که کاربرانی که قصد استفاده از چنین سرویس‌هایی را دارند، پروسه احراز هویت خود را به‌منظور ارتقاء سطح دسترسی و به‌دست آوردن سرویس و استفاده از آن، به‌وسیله ارسال بسته‌های بدون پاسخ از سرور آغاز می‌نمایند [۱۰].

دفاع در عمق یک مدل حفاظتی و لایه‌ای قدرتمند برای اجزاء مهم سیستم‌های اطلاعاتی است، و یکی از روش‌های دفاعی در آن، دفاع لایه‌ای یا چندلایه است. هدف از دفاع لایه‌ای به‌کارگیری از چندین سازوکار تشخیص، نظارتی و حفاظتی است تا یک مهاجم مجبور به عبور از موانع مختلف بازرسی جهت دستیابی به اطلاعات حیاتی گردد. همانطور که در شکل (۱) نشان داده شده است، PKn می‌تواند به‌عنوان اولین لایه دفاع در عمق ایفای نقش نماید. در نتیجه، یکی از مزیت‌های اصلی PKn ایجاد یک لایهٔ اضافی امنیتی علاوه‌بر لایه‌های امنیتی موجود است. بنابراین PKn دارای این قابلیت است که بتواند از لایه چهارم OSI حفاظت نماید [۱۱].



شکل (۱). PKn به‌عنوان اولین لایهٔ دفاعی در استراتژی دفاع در عمق

با توجه به این‌که اکثر سیستم‌های احراز هویت به‌صورت نرم‌افزاری ساخته و پیاده‌سازی شده‌اند و ممکن است در هر لحظه و

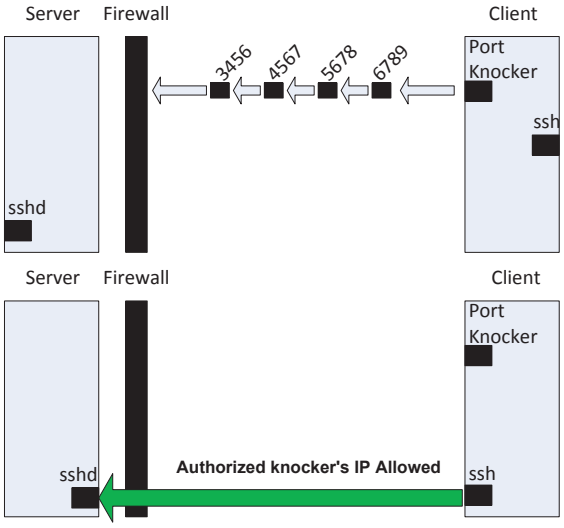
هر زمان آسیب‌پذیری جدیدی در آن‌ها کشف شود و یا این‌که وجود داشته باشند که ما از آن بی اطلاع باشیم (زیرا اکثر آسیب‌پذیری‌های کشف‌شده قبل از اعلان عمومی، مدت‌ها در میان جامعه هکرها دست به‌دست می‌شوند)، بنابراین PKn یک لایه محافظتی بسیار قدرتمندتر نسبت به آن‌ها است.

همچنین اگر از طریق PKn بر روی سرویسی که خود دارای سازوکار احراز هویت درونی است نظیر (SSH)، احرازهویت مجددی صورت گیرد، منجر به آن می‌شود که درصورت عبور از لایهٔ حفاظتی PKn (توسط کاربر مشروع و یا نامشروع)، یک احرازهویت دیگر برای سرویس باقی بماند، که آن احراز هویت ذاتی خود سرویس نظیر احرازهویت موجود در (SSH) است.

به‌این نکته نیز باید توجه داشت که اگر یک پورت برای خود باز باشد (یا شده باشد) هنوز هم مهاجمین به این پورت دسترسی ندارند، زیرا توسط یک بخش امنیتی دیگر (نظیر دیوارهٔ آتش) محافظت شده است. یکی دیگر از مزیت‌های PKn، پنهان‌کاری است؛ دیوارهٔ آتش بر روی سرویس‌دهنده طوری تنظیم شده است که تمام بسته‌ها را رد (Drop) نماید، بنابراین مهاجم نمی‌تواند با اسکن یا کاوش هیچ سرنخی را به‌دست آورد و یا این‌که اصلاً متوجه شود که سروری وجود دارد یا خیر، چه برسد به سرویس‌هایی که بر روی آن در حال اجرا هستند.

شکل (۲) عملکرد یک سیستم PKn را به‌صورت پایه و ساده نشان می‌دهد. به‌طور پیش‌فرض تمامی پورت‌ها بر روی سرور SSHD بسته بوده و تمامی درخواست‌ها در محدوده پورت‌های خاص چه از نوع TCP و یا UDP ثبت گردیده و مابقی نادیده گرفته می‌شوند. روند کار PKn با ارسال یک بسته از سوی مشتری (Client) که حاوی سرآیند IP و سرآیند نوع پروتکل (TCP یا UDP) می‌باشد آغاز می‌شود. درصورتی‌که توالی خاصی از پورت‌ها به‌ترتیب به سرور رسیدند (به‌عنوان مثال به ترتیب ۳۴۵۶، ۴۵۶۷، ۵۶۷۸ و ۶۷۸۹) اقدامات از پیش تعریف‌شده‌ای مانند بازکردن پورت SSH برای میزبانی کلاینت انجام خواهد گرفت و عملیات PKn پایان می‌یابد. حال می‌توان عملیاتی را نیز به‌همین صورت و ترتیب جهت بستن پورت و یا با استفاده از یک زمان‌سنج جهت بستن خودکار پورت (به عنوان مثال بعد از ۳۰ دقیقه) در نظر گرفت.

یکی از مزایای PKn، ایجاد یک اتصال از طریق پورت‌های بسته است. همچنین عملیات Sniffing نیز عملاً غیر ممکن است. با توجه به تعداد پورت‌های TCP که ۶۵۵۳۵ پورت می‌باشد و این‌که اگر و فقط اگر از ۴ توالی Knock مانند شکل (۲) استفاده نماییم، تعداد ۱۸.۴۴۵.۶۱۸.۱۹۹.۵۷۲.۲۵۰.۶۲۵ حالت وجود خواهد داشت، حال کافی است که UDP و یا هر پروتکل دیگری را نیز به توالی خود اضافه نماییم، در نتیجه به یک عدد سرسام‌آور خواهیم رسید که Sniffing را از سوی مهاجمین عملاً غیر ممکن خواهد ساخت. مهم‌تر از همه اینکه PKn را می‌توان بر روی اکثر سیستم عامل‌ها پیاده‌سازی و اجرا نمود.



شکل (۲). دیواره آتش برای سرویس SSH زمانی باز می‌شود که دنباله‌ای از پورت‌های خاص از پیش تعریف‌شده به نظم و ترتیب دریافت

اگرچه PKn باعث می‌شود که فرآیند احرازهویت امن‌تر از قبل گردد، مخصوصاً در شبکه‌هایی که آسیب‌پذیر هستند، با این حال برخی از حملات وجود دارند که بر کارآیی PKn تأثیر گذاشته و امکان سوء استفاده از اتصال‌های صورت‌گرفته را به مهاجمین به‌منظور حمله و تخریب می‌دهند. به برخی از این حملات شناخته شده که عبارت‌اند از DoS-Knocking، Replay Attack، حملات مرد میانی و مشکل موجود در شبکه‌هایی که NAT(network address translation) در آن‌ها صورت گرفته که به مشکل NAT-Knocking معروف می‌باشد می‌توان اشاره نمود. حملات از نوع DoS-Knocking زمانی اتفاق می‌افتند که مهاجمین سیلی از بسته‌های (Packet Flood) تصادفی را به سرور ارسال می‌کنند. در این حالت سرور باید یک بافر برای هر درخواست‌کننده جهت تکمیل باقیمانده فرآیند PKn که از لاگ فایل خوانده می‌شود اختصاص دهد، که این امر منجر به اشغال فضای بسیار زیادی از حافظه گشته و در نتیجه باعث از کار افتادن سرویس می‌گردد [۱۲].

مشکل NAT-Knocking نیز زمانی پدیدار می‌شود که سیستم مانیتورینگ نمی‌تواند کاربر و یا کاربرانی را که احراز هویت شده‌اند از دیگران تشخیص دهد. این سناریو در شبکه‌هایی که از ترجمه آدرس شبکه (NAT) استفاده می‌کنند اتفاق می‌افتد. در نتیجه، تمامی کاربران شبکه محلی با یک آدرس یکسان از خارج شبکه دیده می‌شوند. از این رو وقتی که پروسه PKn یک کاربر به انتها رسید و دسترسی کامل به سرور را پیدا نمود عملاً تمامی کاربرانی که پشت شبکه NAT قرار دارند نیز می‌توانند از این خدمات بهره ببرند [۳ و ۶].

متأسفانه با این‌که PKn به‌عنوان یک سازوکار امنیتی به اندازهٔ کافی دارای مزیت‌های قابل توجهی است اما برخی انتقادها نیز به آن وارد شده است. به‌عبارت دیگر، هنوز جایگاه مناسبی برای PKn در جامعه امنیتی پیدا نشده است. بحث‌های زیادی نیز در خصوص سازوکارهایی که تلاش به دور زدن (bypass کردن) PKn داشته‌اند صورت گرفته است. همچنین PKn متهم به برقراری امنیت از طریق ابهام است، این بدین معنی است که امنیت سیستم از طریق مبهم و مخفی نگه داشتن ویژگی‌های سیستم تأمین گردد. سازوکار PKn از پنهان کاری استفاده می‌کند اما متکی به پنهان‌کاری نیست. به‌صورت مخفیانه باقی‌ماندن، یک رفتار از PKn است، اما هدف اصلی آن نیست. مهاجمین با توجه به سرویس‌های احراز هویت با موقعیت‌یابی و مشاهده دیمن PKn باز هم نمی‌توانند هیچ مزیت مهمی را به‌دست آورند.

۲-۱- تاریخچه PKn

همان‌طور که بیان شد، PKn یک روش بازکردن پورت از خارج از شبکه بر روی دیوارهٔ آتش به‌وسیلهٔ ارسال بسته‌هایی از پیش تعیین‌شده برروی پورت‌های بسته است، که پس از دریافت صحیح توالی، پورت به‌صورت پویا توسط دیوارهٔ آتش پورت باز می‌شود.

یکی از اهداف اصلی PKn جلوگیری از اسکن سیستم و سوءاستفاده و بهره‌برداری از مخاطرات سرویس‌ها، توسط مهاجمین به‌وسیله پورت اسکن بوده که به‌وسیله مخفی نگه داشتن پورت‌ها میسر می‌شود، مگر آنکه نفوذگر دنبالهٔ صحیح Knock را پیدا و سپس ارسال کرده تا پورت بسته‌شده نمایان گردد. براساس وب‌سایت portknocking.org اولین‌بار عبارت Port Knocking توسط M. Krzywinski [13] در سال ۲۰۰۳ ابداع و مطرح گردید.

C.Hammond [۱۴] در وبلاگ خود با نام blog.chipx86.com مدعی این موضوع است که در اواسط سال ۲۰۰۲ ایدهٔ اصلی اضافه کردن یک لایه امنیتی اضافی که توانایی بستن تمام شبکه شخصی خود را داشت و این امکان را نیز برای او فراهم می‌ساخت تا از خارج شهر به آن شبکه دسترسی داشته باشد، اولین بار توسط او مطرح شده و صحت آن را نیز در وبلاگ قدیمی خود در Advogato و پست‌های منتشرشده مربوط به کارهای در حال انجام خود اعلام نموده است، که نام آن را نیز KnockKnock نهاده بود [۱۵]. اما C.Hammond به‌دلیل مشغله کاری فراوان و پروژه‌های موازی نتوانسته بوده ایده خود را به‌موقع به ثبت برساند یا آن را به یک شرکت امنیتی بفروشد. به گفته او، هنگامی‌که در یک روز مجله‌ای را باز کرد که در آن نوشته شده بود Port Knocking ابداع‌شده توسط M.Krzywinski که تماما روش او را پیاده‌سازی کرده بود، برای او بسیار زجرآور بوده، زیرا او این ایده را در سر داشت و یک مقاله نیز دربارهٔ آن نوشته و حتی یک نمونه اولیه از دیمن با نام Knockd (Knockd Daemon) و یک کنسول جهت سرویس گیرنده (Knocker) نیز برای آن آماده کرده بود. روش او همانند کار آقای M.Krzywinski بود و سرویس‌های در حال کاری نظیر ssh, ftp و ... فقط برای knockerهایی که رمز Knock را می‌دانستند باز و برای دیگران بسته نگاه می‌داشت[۱۵].

با این حال اکثر مقالات علمی و مجلات منتشرشده، به مقالات، کارها و سایت M.Krzywinski اشاره و ارجاع داده‌اند و تاکنون اشاره‌ای به نوشته‌های C.Hammond نشده است و شاید دلیل آن اعلام دیر هنگام آن توسط C.Hammond باشد، زیرا این ادعا در مورخ ۱۰فوریه۲۰۱۱ (با توجه به تاریخ ذیل I Invented Port Knocking) در وب سایت blog.chipx86.com اعلام گشته است.

در سال ۲۰۰۲، مقاله‌ای باعنوان تکنیک‌های احرازهویت سبک و پنهان در شبکه‌های مبتنی بر IP توسط P.Barham و همکاران در مرکز تحقیقات اینتل در برکلی ارائه شد [۱۶] که هدف آن‌ها با اهداف PKn شباهت بسیار دارد. در این مقاله سه روش جهت مخفی کردن سرویس‌های TCP/IP از دید مهاجمین و کاربران غیرمجاز معرفی شد. روش‌های ارائه‌شده می‌توانستند در لبهٔ شبکه بدون هیچ تغییری در داخل شبکه راه‌اندازی شده و قادر بودند در کنار پروتکل‌های موجود استقرار یافته و موجب تقویت امنیت آن‌ها شوند.

اما گذشته PKn مربوط به درب‌های پشتی است، که به‌صورت مخفیانه باز می‌شوند. قبل از دیگران دو rootkit با نام‌های cd00r و SAdoor نوعی از Port Knocking را ابداع و

پیاده‌سازی کرده بودند. Cdoor توسط FX از Phenoelit در سال ۲۰۰۰ نوشته شده است [۱۷]، cd00r یک درب پشتی (Backdoor) کنترل از راه‌دور ساده بوده که برای انواع سیستم‌عامل Unix نوشته شده و یک نوع خاص در درب‌های پشتی کنترل از راه‌دور سنتی است. Cd00r به هر پورتی گوش فرا نمی‌دهد بلکه قبل از باز کردن یک پورت برای ارتباطات به تماشای ترافیک IP در میزبانی که در آن ساکن شده است می‌نشیند و به‌دنبال یک توالی خاص از بسته‌ها با ویژگی‌های از پیش تعریف‌شده است. این روش یک Knock مخفی را برای یک کاربر مخرب ایجاد می‌نماید و پس از اینکه Knock مخفی توسط درب پشتی cd00r شناسایی گردید، کاربر مخرب می‌تواند هرگونه کدی را که مایل به آن باشد بر روی میزبان به اجرا درآورد. Knock مخفی می‌تواند مشابه ترافیک عادی و ترافیک پذیرفته‌شده رفتار نماید و این نوع از درب‌های پشتی یا تروجان‌ها می‌توانند چالش منحصر به فردی را برای متخصصان امنیتی سیستم‌های اطلاعاتی که به‌دنبال شناسایی و جلوگیری و محافظت از سیستم‌ها هستند ایجاد نمایند [۱۸]. cd00r با زبان C پیاده‌سازی شده و وقتی Daemon آن بسته‌های TCP SYN خاص با توالی ثابتی از پورت‌ها را شناسایی نماید شروع به‌کار می‌نماید [۱].

SAdoor [۱۱] یک برنامه کامل و تمام‌عیار از نوع درب‌های پشتی غیرقابل شناسایی است که انواع آن از آوریل ۲۰۰۱ تا دسامبر ۲۰۰۳ شناسایی شدند. SAdoor توسط Claes M. Nyberg به زبان C نوشته شده و این برنامه اولین بار برای سیستم‌عامل Unix کامپایل شده بود [۱۹]. درب پشتی SAdoor می‌تواند بر روی سرور به‌عنوان یک جعبه مدیریت از راه دور که در آن فیلترهای pcap (بدون گوش دادن و یا این‌که توسط یک پورت اسکن نمایش داده شوند) به دنبال توالی از پیش تعریف‌شده‌ای از بسته‌های TCP هستند اجرا شود. این توالی یا دنباله می‌تواند شامل پورت‌ها، پرچم‌های TCP و یا هر انتخاب دیگری باشد که از آدرس‌های جعلی آمده باشند. زمانی‌که ترکیب درستی از آدرس مبدأ، پرچم TCP و پورت توسط سرور در حال گوش دادن دریافت شد و توالی با موفقیت توسط کلاینت SAdoor به اجرا درآمد، در آن هنگام سرور آماده دریافت و اجرای دستورات از راه دور است [۲۰]. در ادامه مطالعات و کارهای قبلی صورت‌گرفته بر روی PKn بررسی و ارزیابی می‌شود.

۳- مطالعات و کارهای صورت‌گرفته بر روی PKn

M.Krzywinski در سال ۲۰۰۳ مقاله‌ای در Sys Admin منتشر کرد [۸]. او یک نمونه پایه از PKn را با استفاده از زبان برنامه نویسی پرل طراحی نمود. در این طرح سرویس گیرنده می‌توانست به برنامه در حال اجرای کاملاً ایزوله‌شده که در ابتدا هیچ پورتی بر روی آن باز نبود متصل گردد. این شیوه از یک روش جهت رمز نمودن IP کاربر و قرار دادن آن در دنبالهٔ Knock استفاده کرده و پس از رمزنگاری شدن، بر روی گستره‌ای از پورت‌های میزبان ارسال می‌کرد. این روش، کارآمد و قابل اجرا از PKn بود. از مزایای این روش باز و بسته کردن پویای پورت‌های دیوارهٔ آتش است، یعنی اینکه در یک شبکه ممکن است برنامه‌ها و سرویس‌های مختلفی در حال کار باشند، و در نتیجه هرکدام هروقت که نیاز به دسترسی داشتند باید پورت مربوط به آن باز شود. در این طرح، از IP درخواست دهنده سرویس، پورت، زمان و checksum در توالی Knock استفاده می‌شود. اما این روش در مقابل حملات باز پخش (replay) ناکارآمد بود و هیچ طرحی نیز برای رفع مشکل Nat-Knocking در آن ارائه نشد.

J.B.Ward [۲۱]، در سال ۲۰۰۴ بر اساس طرح اولیه M.Krzywinski که بر پایه بسته‌های TCP کار می‌کرد، پروژه‌ای را با نام Doorman با استفاده از تکنیک PKn پیاده‌سازی نمود. در این روش، قواعد دیواره‌های آتش براساس دستورات صادرشده توسط دیمن Doorman اضافه و یا حذف می‌شود. از مزایای این روش استفاده از UDP و ایجاد حداقل سربار آن است. از معایب این روش این است که جدول های درهم MD5 به‌راحتی در جدول Rainbowشکسته می‌شود و مهاجمین به‌راحتی می‌توانند اطلاعات داخل Knock را به‌دست آورند.

C.K.Tan، مقاله‌ای را با عنوان مدیریت سرور از راه دور با استفاده از PKn پویا و ارسال را جهت از بین بردن مشکل Reply-Attackها یا حملات بازپخش ارائه کرد [۲۲]. یکی از مشکلات PKn بدین صورت است که اگر در سازوکار PKn از توالی‌های ایستا استفاده شود، مهاجمین به‌راحتی قابلیت این را دارند که دنبالهٔ Knock را شناسایی کرده و به پورت و سرویس دسترسی داشته باشند. در این روش نیازی به دنبالهٔ از پیش تعریف شده نبود؛ در عوض، دنباله به‌صورت پویا تعریف‌شده و این باعث می‌شد که از ابهام نیز مبرا گردد. یکی از مزایای این روش چندکاربره بودن آن است و مجبور نبودن توزیع یک توالی بین تمام کاربران است. به عبارت دیگر، در این روش به‌طور مثال به سرویس SSH می‌توان از طریق پورت‌های مختلف در هر زمان دسترسی پیدا کرد که باعث کاهش احتمال حمله و پی بردن دشمن به سرویس SSH می‌شد. در این روش ابتدا سرویس‌گیرنده یک دنباله Knock به‌صورت تصادفی تولید کرده و به‌وسیله یک رمز عبور آن را هش نموده و

سپس به‌صورت یک بسته واحد UDP در قالب MD5 HMAC به‌سمت سرویس دهنده ارسال می‌کند. این روش در ۵ مرحله که هرکدام دارای قوانین مربوط به خود هستند اجرا می‌شود. از معایب این روش پیاده‌سازی بسیار سخت و پیچیده آن است. یکی دیگر از معایب این روش نیاز به نگهداری یک فایل هش از رمزهای عبور تمامی کاربران برروی سرویس دهنده است. در نتیجه باید از این فایل در مقابل مهاجمین محافظت نمود. همچنین این روش در مقابل حملات بازپخش بسیار ضعیف است و باید به‌منظور جلوگیری از این دست حملات، حداقل به آن timestampt اضافه نمود.

D.Worth [۱۱] و همکاران در سال ۲۰۰۴ روشی را با نامه COK (Cryptographic One-Time Knocking) ارائه کرد، او توصیف کرد که استفاده به‌تنهایی از رمزنگاری که IP درخواست‌دهنده به‌عنوان جزئی از محوله Knock قرار دارد در شبکه‌هایی نظیر WiFi Hotspotها و ... که از NAT استفاده می‌کنند موجب باز شدن سرویس برای همه کاربران آن شبکه خواهد شد. D.Worth استفاده از OTP (One-Time Password) را به‌همراه رمزنگاری پیشنهاد کرد. انعطاف‌پذیری OTP به تکرار آن است که براساس یکی از رمزنگاری‌های توابع هش نظیر (SHA1، MD5) کار می‌کند. حملات از نوع بازپخش در تقابل با OTP شکست می‌خورند زیرا امکان استفاده دوباره از یک رمز عبور وجود ندارد. این روش فقط سازگار با دیوارهٔ آتش IPTables است.

درسال‌های ۲۰۰۴ و ۲۰۰۵ مزیت‌ها، قابلیت‌ها و محدودیت‌های موجود در روش‌های ارائه‌شده PKn و مسائلی که به‌طور بالقوه می‌توانستند شبکه‌ها را در معرض خطر قرار دهند مورد بررسی قرار گرفتند [۶ و ۹].

بخش عمده کار بر روی PKn بین سال‌های ۲۰۰۵ تا ۲۰۱۰ صورت گرفت که بیشتر حول محور استفاده از انواع روش‌های رمزنگاری در ارسال توالی Knockها و پیچیده کردن فرآیند آن به‌منظور ارتقاء امنیت بود [۲ و ۲۳]. اما اکثر این روش‌ها منابع بسیار زیادی را از سرویس‌دهنده مصرف می‌کردند و درصورت مواجه شدن با حملات شدیدی از جمله DoS منجر به سرریز بافر شده و عملاً سیستم از کار باز می‌ایستاد.

یکی از نتایج مطالعات انجام شده اخیر، روش Knock به‌صورت بی‌صدا (Silent) است. این روش با استفاده از رمزنگاری بلوکی AES و تابع چکیده‌ساز MD4 منجر به افزایش امنیت گردید. اما نتایج شبیه‌سازی آن نشان داد که این روش باعث ایجاد سربار زیاد و هدر رفتن منابع می‌شود [۲۴] و پیاده‌سازی و اجرای آن بسیار دشوار است.

یکی دیگر از روش‌های احراز هویت امن بر پایه PKn، توسط Liew و همکاران ارائه شد. با توجه به آسیب‌پذیر بودن PKn در حملات NAT-Knocking، این روش با استفاده از ویژگی‌های IPSec و سازوکار رمزهای عبور یکبار مصرف (OTP) ارائه شد. علاوه بر این کارهایی نیز جهت ارتقاء روش‌هایی که از SPA (Single Packet Authentication) استفاده می‌کردند صورت گرفت [۲۶].

در چند سال گذشته کارهایی به‌منظور به چالش کشیدن مفهوم اصلی PKn و مطالب بسیاری به‌منظور ضعیف نشان دادن این روش عنوان شد. با این حال Al-Bahadili و H.Hadi روش PKn هیبریدی را پیشنهاد دادند. در این روش، نشان داده شده است که بسته‌های TCP دارای یک ظرفیت و بار اضافی هستند، که نشان می‌دهد این قبیل سرویس‌ها، به خودی خود باعث افزایش سربار سیستم می‌شوند [۴].

رویکرد دیگری توسط Srivastava و همکاران ارائه گردید، آن‌ها الگوریتمی را پیشنهاد دادند که با استفاده از رمزنگاری AES یک توالی Knock امن را پیاده سازی می‌کرد. در این روش کلاینت درخواست خود را برای OTP از طریق پیامک ارسال و دریافت می‌نمود. علاوه‌براین، آدرس IP مبدأ از یک الگوی مشخص یا توالی از پیش تعریف‌شده تبعیت نمی‌کرد. بنابراین مهاجمین نمی‌توانستند با مانیتورینگ ترافیک، توالی Knock را به‌دست آورند [۵].

روش ساده ضربه‌زدن به پورت در مقابل حملات بازپخش و پویش درگاه توسط F.Ali و همکاران ارائه شد [۲۷]. در این روش به‌جای این‌که پورت‌های مقصد بررسی شوند، پورت‌های بازشده در سمت مبدأ برای ایجاد بستهٔ Knock نیز مورد بررسی قرار می‌گرفتند. در این روش به‌دلیل این‌که تمامی درخواست‌ها فقط به یک پورت ارسال می‌شد امکان ردیابی بسته Knock و طرح‌ریزی حملات بازپخش وجود داشت. در جدول (۱) به‌طور خلاصه عملکرد برخی از طرح‌های PKn به‌همراه نقاط قوت و ضعف آن‌ها بیان شده است.

اکثر روش‌های فوق‌الذکر در مقابل حملاتی نظیر NAT-Knocking، DoS-Knocking، حملات بازپخش (Replay) و یا Snifferها ضعیف بوده و باعث شکست عملیات

جدول (۱). عملکرد، نقاط قوت و ضعف برخی از طرح‌های PKn

عنوان	موضوع	مزایا و معایب
ضربه زدن پایه [۸]	رمز نمودن اطلاعات کاربر (نظیر IP و درگاه) و قرار دادن آن در بسته Knock	نوع پایه از ضربه زدن به درگاه بوده و مستعد مبتلا شدن به حملات بازپخش و متکی به قواعد IPTables است.
بهبود PKn با احراز هویت مستحکم [۲]	استفاده از سیستم احراز هویت چالش و پاسخ	جلوگیری از حملات NAT-Knocking کرده ولی دارای پیاده سازی سخت و دشوار بوده و در مقابل حملات مردمیانی ضعیف است.
Doorman [۲۱].	یک بسته UDPبرای Knock	به‌کارگیری هش MD5 در کنار بسته UDP از نقاط قوت و از معایب آن‌که در جدول Rainbow به‌راحتی MD5 شکسته می‌شود.
یک‌بار ضربه زدن (SPA) [۲۵].	توسعه روش ترکیب PKnو انگشت نگاری غیر فعال سیستم عامل با fwknopبا استفاده از SPA	بسته Knock رمزنگاری شده و بازپخش مجدد آن بسیار دشوار است. مشکل رسیدن خارج از نظم بسته‌ها در این روش مطرح نیست. طرح پیاده‌سازی شده فقط در سیستم عامل لینوکس قابلیت اجرا شدن را دارد.
یک‌بار ضربه زدن (Knocking) با استفاده از SPA و IPSec [۲۶].	استفاده از SPA و OTP با به‌کارگیری از SMS و انجام عملیات PKn به‌منظور دسترسی به تانل IPSec	رمزعبور در بسته PKn بصورت یک‌بار مصرف به‌وسیله سرویس دهندهٔ RNG تولید و به تلفن همراه ارسال می‌شود. ادغام SPA با IPSec و دیوارهٔ آتش بسیار پیچیده و پیاده‌سازی آن بسیار دشوار است.
امنیت شبکه با استفاده از ضربه زدن به درگاه ترکیبی [۴].	ادغام ضربه زدن به درگاه و استگانوگرافی در کناریکدیگر و استفاده از تصاویر جهت Knock	حملات بازپخش در این روش بسیار سخت بوده و این روش بدلیل استفاده توأم از نهانکاری و رمزنگاری دارای سربار بالایی است و اجرای آن در سمت کاربر بسیار پیچیده است.
توسعه معماری کلاینت- سروری PKn با هم‌زمان سازی NTP [۳].	به‌کارگیری از NTPدر جهت ایجاد Knockهایی با طول عمر محدود در مقابل حملات باز پخش	این روش دارای حداقل سربار بوده ولی نیاز به ابزار خاص برای کاربر دارد و نمی‌تواند جلوی حملات بازپخش را گرفته و اینکه در صورت عدم کارکرد و یا دسترسی کلاینت/سرور به سرویس دهنده NTP موجب عدم کارآیی می‌شود.
طرح احرازهویت PKn پیشرفته با QRC با استفاده از AES [۵].	ساخت کلید (یکبار مصرف) جهت الگوریتم رمزنگاری AES برای توالی Knock، به‌همراه رمز مانده مجذوری (QRC)	این طرح به‌دلیل تغییر آدرس IP مبدأ در هر بار Knock برای گمراه نمودن Snifferها بسیار مفید است، اما پیاده‌سازی و اجرای آن بسیار دشوار و کاربر را ملزم به استفاده از برنامه‌های خاص و پیچیده می‌کند.
روش ساده ضربه‌زدن به پورت در مقابل‌حملات بازپخش و پویش درگاه [۲۷].	به‌کارگیری ترکیبی از درگاه‌های مبدأ و مقصد در ارسال و دریافت Knock	این روش از توالی درگاه‌های مبدأ به‌جای توالی درگاه‌های مقصد استفاده نموده است، اما درخواست‌ها فقط به یک درگاه ارسال می‌شوند و امکان حدس زدن و حملات بازپخش در آن وجود دارد.

۴- روش جدید PKn

PKn می‌شدند. در اکثر روش‌های خوب و قدرتمند نیاز به ابزارهای پیچیده و خاص جهت پیاده‌سازی و اجرای PKn بوده و کاربران به‌راحتی نمی‌توانستند در هر کجا و هر زمان از قابلیت‌های PKn استفاده نمایند.

در این مقاله، روش جدیدی که منجر به بهبود امنیت و سهولت اجرای PKn با استفاده از تونل‌زنی با نام SPKT [۲۸]، به‌همراه ویژگی‌های خاص پروتکل کنترل پیام‌های اینترنتی (ICMP) و استفاده از وب (Web) پیشنهاد می‌شود.

بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیامهای اینترنتی، آدرس اینترنتی و تونل زنی روش جدیدی است که در این مقاله ارائه گردیده است. این روش به منظور کاهش حملات بازپخش و ناکارآمد کردن حملات DoS-Knocking و مشکل ساختن Sniffing، ردیابی عملیات و استراق سمع با استفاده از ویژگی‌های خاص ICMP و استفاده همزمان از URLها و قرار دادن توالی Knockها در ICMP و پورت ۸۰ (Web) و همچنین کاهش ریسک امنیتی استفاده در شبکه‌های پشت سرویس NAT

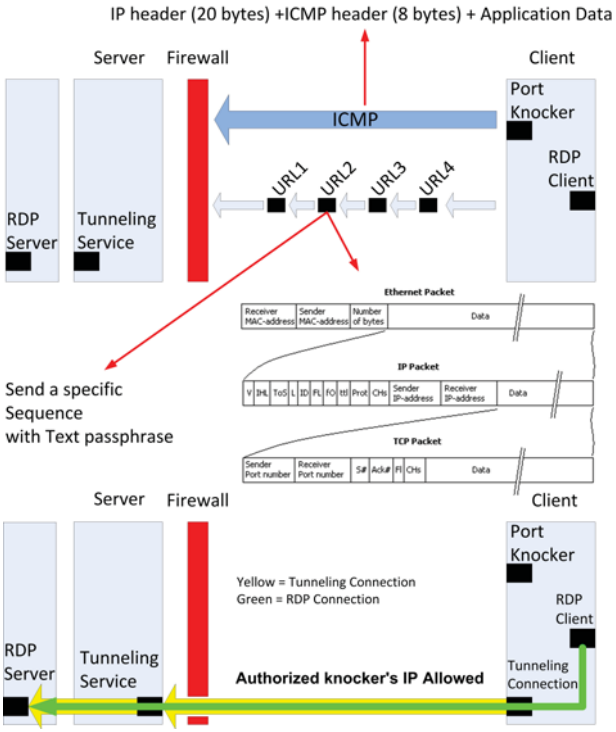
به‌منظور جلوگیری از حملات NAT-Knocking با استفاده از تانلینگ و به‌کارگیری آن با ابزاری ساده و در دسترس در همه‌جا برای استفاده‌کنندگان پیشنهاد می‌شود. این روش را با یک مثال ساده بیان می‌کنیم، شکل (۳)، یک کلاینت را نشان می‌دهد که قصد برقراری ارتباط از راه دور از طریق سرویس RDP را دارد.

به‌طور پیش‌فرض، هیچ پورتی بر روی فایروال باز نیست و تمامی پروتکل‌ها از قبیل TCP، UDP، ICMP و ... بسته هستند، و فایروال تمامی بسته‌ها را رد کرده یا اصطلاحاً Drop می‌نماید. در این زمان دیمن PKn فقط اقدام به بافر نمودن پورت‌های از پیش تعیین‌شده در یک بازهٔ زمانی به‌منظور تکمیل فرآیند PKn می‌نماید.

پس از گذراندن صحیح عملیات و دریافت به‌ترتیب و درست Knockها که ترکیبی از ICMP وURLها هستند، مراحل احراز هویت به روش ارائه‌شده توسط ما به پایان رسیده و سرویس تانل (که می‌تواند سرویس‌هایی نظیر PPTP، L2TP، SSTP و ... باشند) برای IP کاربر (Knocker) فعال شده و سپس پس از برقراری اتصال از طریق تانل، کاربر می‌تواند به سرویس مورد نظر که در این‌جا RDP است، دسترسی پیدا کند. همچنین، می‌توان فایروال را طوری تنظیم نمود که به‌صورت خودکار پس از زمان تعیین‌شده تمامی پورت‌ها را به حال اول (یعنی بسته‌شده) برگرداند و یا این‌که یک سری توالی Knock نیز برای بستن پورت‌ها ارسال نمود. (البته ویژگی استفاده از تانل در این روش اجباری نیست و فقط به‌منظور جلوگیری از حملات NAT-Knocking در شبکه‌هایی که در آن NAT صورت گرفته پیشنهاد می‌شود.) در این روش توالی Knockها علاوه بر ICMP از طریق پروتکل TCP بر روی پورت ۸۰ ارسال می‌گردند. یک بسته پروتکل TCP/IP شامل سرآیند اترنت، سرآیند IP، سرآیند TCP و دیتا می‌شود.

یکی از خصوصیات بارز این پروتکل، تصدیق بسته (ACK) از سوی سرور یا دریافت کننده بسته است. همان‌طور که قبلاً اشاره گردید پاسخگویی و تصدیق بسته‌ها توسط سرور منجر به شناسایی بسته‌ها با استفاده از عملیات Sniffing شده و می‌تواند باعث حملاتی مانند Replay Attackها و یا دیگر حملات از نوع مرد میانی (Man-In-The-Middle-Attack) گردد.

با توجه به اینکه هیچ سرویسی در سمت سرور در حال اجرا و یا پورتی باز نیست، در نتیجه هیچ بسته تصدیقی (ACK) از سوی سرور به سمت کاربر ارسال نمی‌شود و تمامی حملاتی که از طریق بسته تصدیق پی ریزی و اجرا می‌شوند کاملاً بی‌اثر خواهند شد.



شکل (۳). بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام‌های اینترنتی، آدرس اینترنتی و تونل زنی

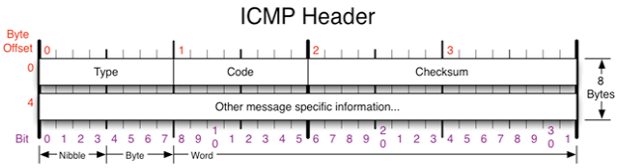
در روش‌های مرسوم PKn مانند شکل (۲)، کلاینت فقط به ارسال بسته‌های Knock با یک توالی منظم اقدام می‌کرد، اما در این روش به‌منظور امنیت بیشتر، به‌همراه بسته‌های TCP (بر روی پورت ۸۰) که عمل Knock را انجام می‌دهند، یک رشته متن به‌صورت URL به‌عنوان کلمه عبور (passphrase) نیز در داخل بسته‌های http جای‌گذاری شده و برروی پورت ۸۰ به سرور ارسال می‌شود.

اما مزیت این روش نسبت به SPK [۱۱]، به‌کارگیری از صفحات وب و استفاده از مرورگر به‌عنوان ابزار ارسال Knock و پروتکل کنترل پیام‌های اینترنتی یا ICMP و استفاده از برنامه‌ای نظیر Pingبه عنوان ابزاری دیگر جهت ارسال Knock، به‌منظور افزایش امنیت و استفاده سهل و آسان از این ابزارها در حین عملیات Knock است.

مزیت این روش نسبت به PWIT [۲۹] استفاده از ویژگی‌های خاص موجود در پروتکل کنترل پیام‌های اینترنتی (ICMP) و ابزارهای آن نظیر Ping است. این ویژگی‌ها Timestamp و Time To Live هستند که در این روش مورد استفاده قرار گرفته‌اند.

Timestamp در ICMP یکی از ویژگی‌های خاص موجود در دستور ping بوده که بسیار کم از آن استفاده می‌شود. Timestamp در واقع ثانیه‌های سپری‌شده از اول ژانویه ۱۹۷۰ (۱۹۷۰/۰۱/۰۱ ۰۰:۰۰:۰۰) تا زمان حاضر است و زمان حاصل از آن یک عدد صحیح خواهد بود. گزینه Timestamp در ابزار Ping، در سرآیند IP برای ثبت زمان ورود درخواست پیام Echo و پاسخ پیام Echo مورد استفاده قرار می‌گیرد.

Time To Live یا TTL از در حلقه افتادن بسته‌های Ping جلوگیری می‌کند، TTL تعداد Hopها را در مسیر خود شمارش



شکل (۴). سرآیند پروتکل کنترل پیام‌های اینترنتی (ICMP) [۳۰]

کرده و در هر Hop یک شماره از TTL کم می‌شود. وقتی که عدد TTL به صفر برسد، این بدان معناست که زمان تعیین‌شده تمام شده است.

در این روش ابتدا بسته ICMP با یک حجم مشخص به‌همراه Timestamp و TTL ارسال می‌شود، در صورتی‌که بسته ICMP با همان سایز ارسالی، Timestamp و TTL توسط دیواره آتش دریافت گردید، امکان ارسال بسته‌های Knock به‌صورت URL میسر می‌شود.

به این نکته نیز باید توجه داشت که سایز بسته ارسالی توسط کاربر و سایز بسته ICMP دریافتی در سمت سرور (دیوارهٔ آتش) باهم برابر نیستند. به‌طور مثال اگر کلاینت بسته ICMP را با سایز ۱۰۰ بایت ارسال نماید و در سمت دیواره‌آتش و سرویس Port-Knocking شرط دریافت بسته ICMP و عملیات Knock نیز، ۱۰۰ بایت در نظر گرفته شده باشد، هرگز چنین شرط و عملیاتی برقرار نخواهد شد.

همان‌طور که در شکل (۴) نشان داده شده است، اندازه سرآیند یک بسته ICMP، هشت بایت است، در نتیجه این اندازه نیز به اندازهٔ بسته ارسالی ICMP کلاینت اضافه می‌گردد.

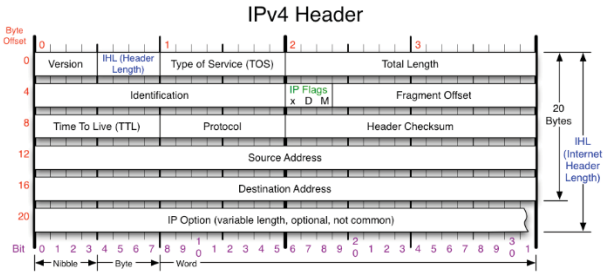
ICMP Header (8 Bytes) + Packet-Size

همچنین علاوه بر سرآیند ICMP، سرآیند IP نیز به بسته ارسالی اضافه می‌شود. همانگونه که در شکل (۵) نشان داده شده است، سرآیند IP نسخه ۴، بیست بایت می‌باشد؛ در نتیجه، اندازه بسته ارسالی به‌صورت زیر محاسبه می‌گردد :

IP Header (20 Bytes) + ICMP Header (8 Bytes) + Packet-Size

و اگر طبق مثال بالا، کلاینت ۱۰۰ بایت را در یک بسته ارسال نماید، در سمت دیوارهٔ آتش، آن بسته با اندازه ۱۲۸ بایت دریافت می‌شود، بنابراین شرط دریافت این نوع پروتکل باید ۲۸ بایت بیشتر از سایز بسته کلاینت در سمت سرور تنظیم گردد.

همچنین مقدار TTL تنظیم‌شده در درخواست پیام Echo با پاسخ پیام Echo متفاوت است، زیرا مقدار پیش‌فرض TTL هر سیستم عامل در پاسخ پیام Echo برگشت می‌نماید. به‌طور مثال اگر مقدار TTL را در ارسال بسته ICMP به یک سیستم عامل MikroTik بر روی عدد ۱۰۰ تنظیم گردد، مقدار TTL در پاسخ پیام Echo برابر ۶۴ خواهد بود. در جدول (۲) مقادیر اولیه TTL برخی از سیستم‌عامل‌ها نشان داده شده است.



شکل (۵). سرآیند IPv4 [30]

شایان ذکر است از مقدار TTL و TCP Window Size برای شناسایی نوع سیستم عامل استفاده می‌شود. به این روش، انگشت‌نگاری از سیستم عامل (OS Fingerpringing) می‌گویند. این مقادیر را می‌توان بسادگی با مانیتور کردن ترافیک در اولین بسته یک جلسه TCP در سرآیند IP مشاهده نمود. دلیل اصلی این‌که سیستم عامل‌ها دارای مقادیر یکسانی نبوده و متفاوت هستند در این نکته نهفته است که در RFC مربوط به TCP/IP مقادیر پیش‌فرضی در این خصوص قید نشده است و هر سیستم عامل عدد مخصوص به خود را دارد.

نکته مهم در این خصوص این است که مقدار TTL همیشه با مقادیر جدول (۲) مطابقت ندارند. به‌طور مثال اگر در حال استفاده از

سیستم‌عامل ویندوز ۷ باشیم و هنگامی‌که که یک بسته IP بـر روی شبکه به سمت سیستم عامل ما ارسال شود مقدار TTL بـه‌دلـیـل عبور از مسیریاب‌ها به ازای هر مسیریاب یک واحد کاهش می‌یـابـد. بنابراین اگر مقدار TTL بسته برابر با ۱۱۶ مشاهده شد اینطـور در نظر گرفته می‌شود که TTL برابر با ۱۲۸ بوده و بسـتـه از ۱۲ روتـر عبور نموده است.

جدول (۲). مقادیر اولیه TTL و TCP Window Size در برخی از سیستم‌عامل‌ها

Operating System (OS)	IP Initial TTL	TCP Window Size
Linux (kernel 2.4 and 2.6)	۶۴	۵۸۴۰
Google's customized Linux	۶۴	۵۷۲۰
FreeBSD	64	۶۵۵۳۵
Windows XP	۱۲۸	۶۵۵۳۵
Windows 7 & Server 2008	۱۲۸	۸۱۹۲
Cisco Router (IOS 12.4)	۲۵۵	۴۱۲۸

پس از این‌که بسته ICMP، به‌عنوان اولین Knock با انـدازه و TTL مشخص شده به‌همراه Timestamp توسـط دیـوارهٔ آتـش تأیید گردید، در واقع فاز اول این روش تکمیـل گـردیـده و سـپـس کلاینت می‌تواند اقدام به ارسال توالی Knockها بـه‌صـورت URL نماید. درنوع مرسوم Port-Knocking، پس از آن‌که عمل ارسـال بسته‌های دنبالهٔ Knock به پـایـان مـی‌رسـیـد، ارتـبـاط بـرقـرار (Established) می‌گردید. اما در این روش به‌همراه دنبالهٔ ارسالی، متن کلمهٔ عبور نیز باید به‌همراه هر Knock ارسال گردد، تا فاز دوم این روش تکمیل گردد.

این دنباله به‌جای این‌که به پورت‌های مختلف ارسال گردد فقـط به یک پورت یعنی پورت ۸۰ ارسال می‌شود و در هـر بـار Knock آدرس‌های URL تغییر می‌کند. حال پس از انجام مراحل ذکرشـده وارد فاز سوم و فاز نهایی این روش یعنی بخش تانل می‌شویم. در این بخش دیوارهٔ آتش پس از تأیید مراحل قبل، اجازهٔ دسترسی به تانـل مربوطه (نظیر PPTP بر روی پروتکل GRE) را به کلاینت داده تـا امکان ایجاد ارتباط بر روی تانل برای کلاینت مـیسر گـردد. پـس از ایجاد ارتباط بر روی تانل، همان‌طور که در شکل (۳) نشان داده شده است کلاینت می‌تواند با سرویس مورد نظر (مانند سـرویـس RDP) ارتباط برقرار نماید.

البته شایان ذکر است که تانل یک بخش اختیاری اسـت و بـه‌منظور جلوگیری از حملات NAT-Knocking در شبکه‌هایی که سرویس‌گیرنده در پشت شبکه NAT قرار دارد مفیـد اسـت و اگـر سرویس‌گیرنده پشت شبکه NAT قرار نداشته باشد و دارای یک IP معتبر باشد نیاز به اجرای این قسمت نیست. اما مزیت اسـتـفـاده از تانل حتی در شبکه‌ای که در آن NAT صـورت نـگـرفـتـه بـاشـد، جلوگیری از Sniffing و سرقت اطلاعات است زیـرا بسـتـه‌هـا در داخل یک تانل و به‌صورت امن تبادل می‌شوند.

در شکل (۶) یک مثال از روش جدیـد ارائـه‌شـده نشـان داده شده است. دنباله PKn پس از پنج بار Knock تکمیل می‌شود. در این مثال یک کلاینت با IP آدرس ۱۱۱.۱۱۱.۱۱۱.۱۱۱ شـروع بـه ارسال بسته ICMP با سایز ۱۰۰ بایت بـا TTL بـا انـدازهٔ ۴۰ و Timestamp به سرور می‌نماید.

سپس درصورتی‌که سایز بسته دریافتی توسط سرور، ۱۲۸ بایت، اندازهٔ TTL برابر ۴۰ و Timestamp در بسته موجود بود، سرور نسبت به بافر کردن اطلاعات IP (آدرس کلاینت) به مدت ۴۰ ثانیـه در لیست موقت که در این مثال temprory1 نامیـده شـده اسـت، اقدام می‌کند.

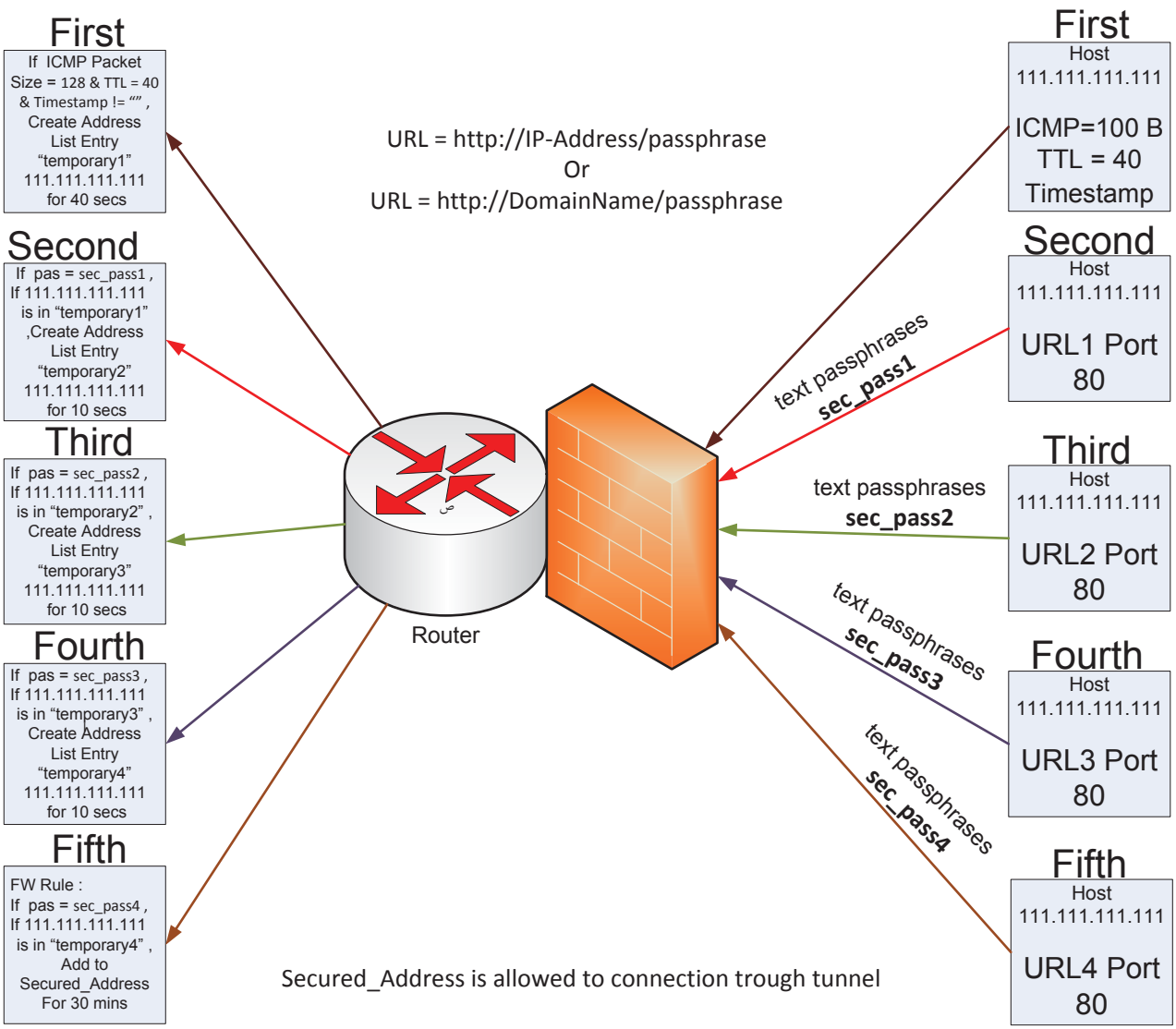
کلاینت، ۴۰ ثانیه فرصت دارد تا Knockهای بعدی را انـجـام دهد در غیر این‌صورت اطلاعات او از لیسـت مـوقـتtemproray1 به‌صورت خودکار حذف خواهد شد.

همچنین اگر کلاینت اطلاعاتی را غیر از اطلاعات اصلی به سـرور ارسال نماید، سرور هیچ حافظه‌ای برای آن تخصیص نخـواهـد داد و تمامی بسته‌ها توسط سرور دور ریخته شده یـا اصـطـلاحـاً Drop می‌شوند. بنابراین مشکل DoS-Knocking رخ نخواهد داد.

طبق مثال، در مرحله بعد کلاینت آدرس‌هـای URL را بـرروی پورت شماره ۸۰ به‌صورت ذیل با استفاده از مرورگر ارسال می‌نماید:

http://server-ip/passphrase

سرور معتبر بودن پورت را بررسی کرده و درصـورتـی‌کـه IP آدرس کلاینت در لیست temproray1 وجود داشت و کلمه عبـور نیز برابر بود نسبت به ذخیره اطلاعات در لـیـسـت temproray2 اقدام می‌نماید. در مراحل بعدی Knock نیز، PKn علاوه بر بررسی IP آدرس در لیست موقت، کلمه عبور را نیز بررسی نموده و این کار را برای چهار مرحله انجام می‌دهد.



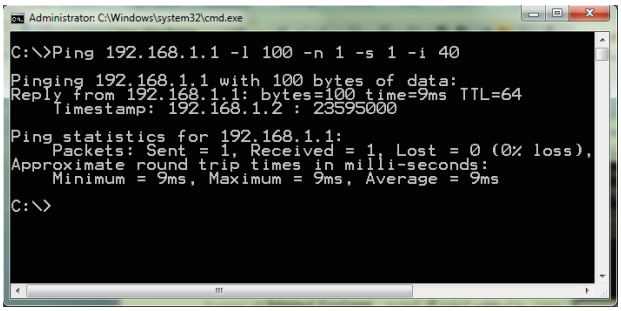
شکل (۶). مراحل روش جدید ارائه‌شده

کل فرآیند باید در ۴۰ ثانیه تکمیل گردد، اگر هریک از بسته‌های URL در زمان ۱۰ ثانیه نتوانند به سرور برسند، به‌صورت خودکار، بافر خالی گشته و پروسه باید دوباره آغاز گردد. درصورتی‌که مراحل PKn با موفقیت به پایان رسد، فاز بعدی که فراهم شدن امکان تانل به سرور توسط کلاینت است، آغاز به‌کار خواهد نمود.

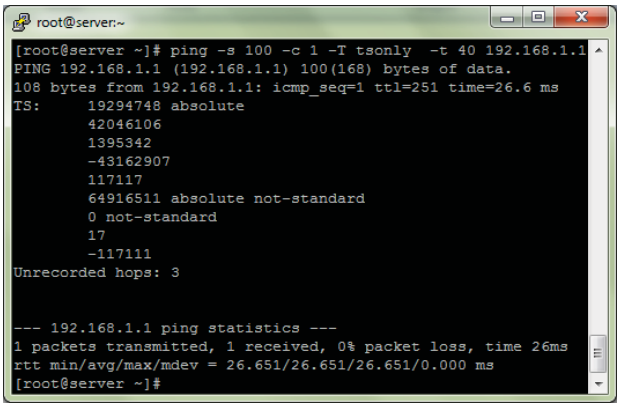
در این بخش، پس از تکمیل مراحل احراز هویت PKn، امکان دسترسی به سرویس تانل به IP کلاینت داده شده است و کلاینت می‌تواند با ایجاد یک اتصال از طریق تانل به سرویس RDP متصل شود (شایان ذکر است سرویس تانل فقط به IPهای تانل پاسخ می‌دهد). هر جلسه کاری به‌مدت ۳۰ دقیقه باز خواهد بود و به‌صورت خودکار بسته خواهد شد.

۵- نتایج پیاده‌سازی

تکنیک جدید PKn معرفی‌شده (بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام‌های اینترنتی، آدرس اینترنتی و تونل‌زنی) ، برروی سیستم‌عامل RouterOS روتربردهای سری ۱۱۰۰ میکروتیک با هسته Linux 3.3.5 مورد بررسی و پیاده‌سازی قرار گرفته است. اسکریپت‌های استفاده‌شده جهت پیاده‌سازی این روش در سمت سرور در بخش ضمیمه ارائه شده است.



شکل (۷). ارسال بسته ICMP در ویندوز ۷



شکل (۸). ارسال بسته ICMP در CentOS Linux 6.3

۵-۱- کلاینت

به‌منظور پیاده‌سازی در سمت کلاینت از سیستم عامل ویندوز ۷ و لینوکس CentOS 6.3 برای ارسال پروسه بسته‌های Knock استفاده شد (البته می‌توان از انواع مختلف سیستم عامل مانند انواع نسخه‌های لینوکس و ... نیز استفاده نمود)، در ابتدا با استفاده از دستور Ping ارسال بسته ICMP به سمت سرور صورت گرفت، شکل‌های (۷ و ۸).

در آزمون صورت‌گرفته سایز بسته ICMP، ۱۰۰بایت و مقدار TTLنیز ۴۰ در نظر گرفته شده است. و با دستور زیر در ویندوز بسته به سمت سرور ارسال می‌شود (IP سرور ۱۹۲.۱۶۸.۱.۱ در نظر گرفته شده است) :

Ping 192.168.1.1 –l 100 –n 1 –s 1 –i 40

راهنمای گزینه‌های استفاده‌شده بدین شرح است :

l- : اندازهٔ بافر ارسالی را مشخص می‌کند.

n- : تعداد درخواست‌های echo برای ارسال را مشخص می‌کند.

s-: timestamp را مشخص می‌سازد که می‌تواند بین ۱ تا ۴ باشد.

i-: مقدار Time to Live را مشخص می‌سازد.

همچنین اگر در لینوکس نیز بخواهیم بسته‌های ICMP را ارسال نماییم شکل دستور به‌صورت زیرخواهد بود :

ping -s 100 -c 1 -T tsonly -t 40 192.168.1.1

s-: اندازهٔ بافر ارسالی را مشخص می‌کند.

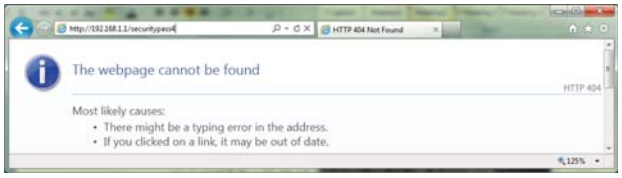
c-: تعداد درخواست‌های echo برای ارسال را مشخص می‌کند.

T-: timestamp را مشخص می‌سازد که می‌تواند tsonly (فقط Timestamp و یا tsandaddr (Timestamp به‌همراه آدرس IP باشد.

t-: مقدار Time to Live را مشخص می‌سازد.

سپس با استفاده از مرورگر اینترنتی Internet Explorer(در ویندوز) و یا هر مرورگر اینترنتی دیگر، آدرس‌ها به‌همراه رمزهای عبور به‌صورت (URL) به ترتیب از پیش تعیین‌شده به‌عنوان دنباله Knockها ارسال شدند. شایان ذکر است اگر سایز بسته ارسالی ICMP، ۱۲۸ بایت باشد (شرط باید به‌دلیل سرآیند IPv4 و ICMP تعداد ۲۸ بایت بیشتر است)، سرور به ارسال توالی URLها واکنش نشان می‌دهد.

به‌دلیل این‌که هیچ سرویس‌دهندهٔ وبی در آدرس سرور، در حال سرویس‌دهی نیست، در نتیجه هیچ پاسخی دریافت نگردیده و پیغام "The webpage cannot be found"و یا خطای معروف ۴۰۴ صادر و هیچ اتصالی نیز صورت نمی‌گیرد، شکل (۹). همچنین مانیتور کردن شبکه و اتصال‌ها توسط مهاجمین به‌منظور دست‌یافتن به فرآیند Knock بسیار سخت و دشوار است، این فرآیند نیز با نرم‌افزار Wireshark بررسی گردید.



شکل (۹). ارسال URLها

۵- ۲- سرور

جهت جستجوی الگوی رمزهای عبور ارسالی (passphrase) در سمت سرور از Layer7-Protocol که یک روش برای جستجوی الگوهای موجود در KB ۲ اول یا ۱۰ بسته اول هر اتصال است استفاده گردید. اسکریپت هایی نیز به‌منظور جلوگیری از حملات DOS استفاده شد. در این اسکریپت‌ها حداکثر اتصال براساس IP، صد اتصال هم‌زمان درنظر گرفته شده است. اسکریپت های ارائه‌شده در بخش ضمیمه با استفاده از محدود نمودن تعداد اتصال براساس هر آدرس مبدأ می‌توانند جلوی حملات DOS را گرفته و همچنین چون هیچ پورتی در سمت سرور بدین منظور باز نیست عملاً حملات DOS-Knocking تأثیری ندارند.

از نرم‌افزار معروف nmap به‌منظور عملیات جستجوی پورت باز استفاده شد، که نتیجه کار، هیچ پورت بازی را بر روی سرور نشان نداد و اینکه مشکل NAT-Knocking با استفاده از فعال‌سازی تانل به حداقل رسانده شده است و اگر کلاینت در یک شبکه در پشت سرویس NAT قرار داشته باشد، نیاز به احراز هویت مجدد جهت ایجاد ارتباط تانل، این مشکل را نیز برطرف خواهد نمود.

۶- نتیجه‌گیری

در تجزیه و تحلیل و بررسی روش‌های مختلف PKn مشخص شده است که برخی مشکلات در طراحی و پیاده‌سازی، امکان دسترسی به کاربران غیرمجاز را می دهند و یا منجر به از کارافتادن سرویس می‌شوند، همچنین به ابزارهای خاص و یا پیچیده جهت پیاده‌سازی و اجرا نیاز داشتند. بهره‌گیری از Port-Knocking به‌عنوان اولین لایه دفاعی در استراتژی دفاع در عمق با استفاده ترکیبی از ویژگی‌های پروتکل کنترل پیام‌های اینترنتی، آدرس اینترنتی و تونل‌زنی، یک روش نوین در حوزهٔ PKn بوده که در این مقاله ارائه گردیده و منجر به بهبود احرازهویت به روش Port-Knocking شده است. این روش به‌دلیل سهولت اجرا در سمت سرویس‌گیرنده، کلاینت را بی‌نیاز از ابزارهای خاص و پیچیده می‌کند و با استفاده از ابزارهای موجود در OSها (مرورگرها و Ping) به‌راحتی قابلیت اجرا در هرکجا و هرزمان را دارد. این روش علی‌رغم سادگی در بخش کلاینت در بخش سرور بسیار سبک اما پرقدرت است. استفاده همزمان از ویژگی‌های خاص ICMP و صفحات وب که از پورت‌ها و یا پروتکل‌های غیر متعارف استفاده نمی‌کنند و بطور پیش‌فرض نیز پورت ۸۰ (مربوط به وب سرورها) و تا حدودی هم ICMP دارای ترافیک قابل ملاحظه و بالایی در شبکه‌ها و اینترنت هستند، در نتیجه مشکلات مربوط به Sniffing و حملات مرد میانی را به حداقل رسانده است. این روش در مقابل حملات بازپخش (Replay) که درصورت موفقیت در Sniffing صورت می‌گیرد، ضعیف بوده اما می‌توان با به‌کارگیری از OTP (رمزهای یک‌بار مصرف) و یا Timestampt و گنجاندن آن‌ها در بسته‌های ارسالی این مشکل را نیز حل نمود.

همچنین چون هیچ پورتی بر روی سرور بدین منظور باز نبوده و سرویس دهی عمومی نمی نماید عملاً حملاتی از نوع DOS -Knocking را بی‌اثر کرده و استفاده از تانل مشکلات مربوط به NAT-Knocking را به حداقل رسانده است و امنیت اتصال به سرویس‌ها را به‌دلیل استفاده از تانلینگ تضمین نموده و به‌راحتی قابلیت سفارشی کردن را داشته و می‌توان مراحل Knock را کم یا زیاد نمود. از مهم‌ترین کارهایی که می توان در آینده بر روی این روش انجام داد، کنترل تعداد جلسات برای ارتباطات و استفاده از OTP، Timestampt و SMS است که منجر به افزایش سطح امنیت می‌شوند.

۷- مراجع

[1] M. Krzywinski, "Port knocking from the inside out," Communication, 2005.

[2] R. deGraaf, J. Aycock, and M. J. Jacobson, "Improved Port Knocking with Strong Authentication," 21st Annu. Comput. Secur. Appl. Conf., no. Acsac, pp. 451–462, 2005.

[3] T. Popeea, V. Olteanu, L. Gheorghe, and R. Rughinis, "Extension of a port knocking client-server architecture with NTP synchronization," in Roedunet International Conference (RoEduNet), 2011 10th, pp. 1–5, 2011.

[4] H. Al-Bahadili and A. H. Hadi, "Network Security Using Hybrid Port Knocking," IJCSNS, vol. 10, no. 8, pp. 8, 2010.

[5] V. Srivastava, A. K. Keshri, A. D. Roy, V. K. Chaurasiya, and R. Gupta, "Advanced port knocking authentication scheme with QRC using AES," in Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on, pp. 159–163, 2011.

[6] B. Maddock, "Port Knocking: An Overview of Concepts, Issues and Implementations," SANS GIAC GSEC Pract. 23rd, 2004.

[7] M. Krzywinski, "Port Knocking," A System for

Stealthy Authentication Across Closed Ports, Available: http://www.portknocking.org/view/about/summary, 10-Dec-2012.

[8] M. Krzywinski, "Port Knocking - Network Authentication Across Closed Ports," SysAdmin, vol. 12, no. 6, pp. 12–17, 2003.

[9] S. Jeanquier, "An Analysis of Port Knocking and Single Packet Authorization MSc Thesis," 2006.

[10] D. Worth, "COK: Cryptographic one-time knocking," Talk slides, Black Hat USA, 2004.

[11] OSI Reference Model, [Online]. Available: http://standards.iso.org/ittf/Publicly Available Standards/index.html, 24-Nov-2012.

[12] A. I. Manzanares, J. T. Marquez, J. M. Estevez-Tapiador, and J. C. H. Castro, "Attacks on port knocking authentication mechanism," Comput. Sci. Its Appl. 2005, pp. 1292–1300, 2005.

[13] M. Krzywinski, "Port Knocking," [Online]. Available: http://www.portknocking.org/, 12-Dec-2012.

[14] C. Hammond, "I Invented Port Knocking," 2011. [Online]. Available: http://blog . Chipx86 .com/2011/02/10/i-invented-port-knocking/, 24-Dec-2012.

[15] C. Hammond, "Knock Knock," [Online]. Available: http://www.advogato.org/person/chipx86/diary/134.html, 24-Dec-2012.

[16] P. Barham, S. Hand, and R. Isaacs, "Techniques for lightweight concealment and authentication in IP networks," Intel Res. Berkeley, 2002.

[17] F. of Phenoelit, "cd00r.c," 2000. [Online]. Available:http://www.phenoelit.org/stuff/d00rdescr.html, 22-Dec-2012.

[18] G. Hartrell, "Get ahandle oncd00r: The invisible backdoor," SANS Inst., no. Security 504, 2002.

[19] C. M. Nyberg, "SAdoor," 2001. [Online]. Available: http://cmn.listprojects.darklab.org/. 10-Jan-2013.

[20] Creining, "Undetectable backdoor SAdoor," 2003. [Online]. Available: http://packetfu.org/2003/04/undetectable-backdoor-sadoor.html, 10-Jan-2013.

[21] J.B.Ward, "The Doorman or Silent Running," 2004 [Online], Available: http://doorman. source-forge.net/.

[22] C. K. T. Cappella, "Remote Server Management Using Dynamic Port Knocking and Forwarding," Security, 2004.

[23] P. Iyappan, K. S. Arvind, N. Geetha, and S. Vanitha, "Pluggable Encryption Algorithm In Secure Shell(SSH) Protocol," 2009 Second Int. Conf. Emerg. Trends Eng. Technol., pp. 808–813, 2009.

[24] E. Y. Vasserman, N. Hopper, and J. Tyra, "SilentKnock: practical, provably undetectable authentication," Int. J. Inf. Secur., vol. 8, no. 2, pp. 121–135, Nov. 2009.

[25] M. Rash, "Advances In Single Packet Authorization," ShmooCon, no. 2, 2006.

[26] J. H. Liew, S. Lee, I. Ong, H. J. Lee, and H. Lim, "One-Time Knocking framework using SPA and IPsec," in Education Technology and Computer (ICETC), 2010 2nd International Conference on, vol. 5, pp. 205–209, 2010.

[27] F. Ali, R. Yunos, and M. Alias, "Simple port knocking method: Against TCP replay attack and port scanning," Cyber Secur. Cyber Warf., pp. 247 –252, Jun. 2012.

[28] P. Mehran, E. A. Reza, and B. Laleh, "SPKT: Secure Port Knock-Tunneling, an enhanced port security authentication mechanism," 2012 IEEE Symp. Comput. Informatics, pp. 145–149, Mar. 2012.

[29] M. Pourvahab, R. E. Atani, and M. Shavanddasht, "Port-Knocking with the usage of Web, Internet control message protocol and Tunneling (PWIT)," in 6th National Conference Iran"s Scientific Society on Command, Control, Communications, Computer and Inteligence (C4I), 2012.

[30] M. Baxter, "TCP/IP Reference," [Online] Available: http://nmap.org/book/tcpip-ref.html, Accessed: 10-Apr-2013.

۸- ضمیمه

اسکریپت‌های سمت سرور جهت بررسی Knockها

```
/ip firewall filter

add action=add-src-to-address-list address-list=temporary1 address-list-timeout=10s chain=input comment="ICMP - 128 Bytes" disabled= no packet-size=128 protocol=icmp ipv4-options=timestamp ttl=equal:40

add action=drop chain=input disabled=no dst-port=80 protocol=tcp src-address-list=!temporary1

add action=add-src-to-address-list address-list=temporary2 address-list-timeout=10s chain=input comment="URL1" disabled=no dst-port=80 layer7-protocol=knock_url1 protocol=tcp src-address-list= temporary1

add action=add-src-to-address-list address-list=temporary3address-list-timeout=10s chain=input comment="URL2" disabled=no dst-port=80 layer7-protocol=knock_url2 protocol=tcp src-address-list=temporary2

add action=add-src-to-address-list address-list=temporary4 address-list-timeout=10s chain=input comment="URL3" disabled=no dst-port=80 layer7-protocol=knock_url3 protocol=tcp src-address-list=temporary3

add action=add-src-to-address-list address-list="Temp allow" address-list-timeout=10m chain=input comment="URL4" disabled=no dst-port=80 layer7-protocol=knock_url4 protocol=tcp src-address-list=temporary4

add action=accept chain=input comment=Allow disabled=no protocol=gre src-address-list=Temp_allow

add action=drop chain=input comment="Everything Connection" disabled=no dst-port=3389 !src-address-list=Secured_address protocol=tcp
```

اسکریپت‌های سمت سرور جهت جلوگیری حملات DOS

```
/ip firewall filter add chain=input protocol=tcp connection-limit=100,32 action=add-src-to-address-list address-list=blocked-addr address-list-timeout=1d
/ip firewall filter add chain=input protocol=tcp src-address-list=blocked-addr connection-limit=3,32 action=tarpit
/ip firewall filter add chain=forward protocol=tcp tcp-flags=syn connection-state=new action=jump jump-target=SYN-Protect comment="SYN Flood protect" disabled=yes
/ip firewall filter add chain=SYN-Protect protocol=tcp tcp-flags=syn limit=400,5 connection-state=new action=accept comment="" disabled=no
/ip firewall filter add chain=SYN-Protect protocol=tcp tcp-flags=syn connection-state=new action=drop comment="" disabled=no
/ip firewall connection tracking set tcp-syncookie=yes
```

اسکریپت‌های بررسی الگوی رمزعبور در بسته‌های پورت ۸۰ با استفاده از Layer7-Protocol

```
/ip firewall layer7-protocol
add name=knock1 regexp=securitypass1
add name=knock2 regexp=securitypass2
add name=knock3 regexp=securitypass3
add name=knock4 regexp=securitypass4
add name=HTTP regexp="http/(0\\.9|1\\.0|1\\.1) [1-5][0-9][0-9] [\\x09-\\x0d\~-
)*[ connection:|content-type:|content-length:|date:)|post [\\x09-\\x0d\*[~-
_\ http/[01]\\.[019"[
add name=HTTP1 regexp="http/(0\\.9|1\\.0|1\\.1) [1-5][0-9][0-9]|post [\\x09\\\-
x0d-~]* http/[01]\\.[019"[
add name=HTTP2 regexp="(http[\\x09-\\x0d-~]*(200 ok|302|304 )[\\x09-\\x0d\~-
)*[ connection:|content-type:|content-length:))|^(post [\\x09-\\x0d-~]* ht\
tp"(/
```