امنیت در وب

حمیدرضا نیرومند

niroomand.ir

زمستان ۹۴

فهرست مطالب

فصل اول

۱-۱ امنیت چیست؟

امنیت به معنی حفاظت از داشته‌هاست. این داشته‌ها می‌تواند صفحات وب و یا دیتابیس مشتری باشد و با حتی حفاظت از شهرت شرکت. امنیت یک مسیر است و نه یک مقصد[۱]. همینطور که زیرساخت و برنامه‌تان را آنالیز می‌کنید، تهدیدات بالقوه‌ای را شناسایی می‌کنید که هر کدام درجه‌ای از ریسک را در پی دارد. امنیت در حقیقت مدیریت ریسک و ارائه‌ی یک اقدام مقابل و موثر است.

۱-۱-۱ عناصر اصلی امنیت(پایه‌های امنیت) :

Authentication[2] 1-1-1-1: به این سوال پاسخ می‌دهد: Who are you? (شما کی هستید؟) در این فرآیند سرویس‌گیرنده‌های برنامه‌ها و سرویس‌های شما به صورت یکتا شناسایی می‌شود. افراد احرازهویت شده ممکن است کاربران نهایی[۳]، سرویس‌ها، فرآیندها یا کامپیوترهای دیگر باشند. در اصطلاحات مربوط به امنیت به سرویس‌گیرنده‌های احرازهویت شده در اصطلاح Principals[4] گفته می‌شود.

Authorization[5] 2-1-1-1: به این سوال پاسخ می‌دهد که What can you do? (شما چه کاری می‌توانید انجام دهید؟) این فرآیند تعیین می‌کند که کاربران احرازهویت شده به چه منابع و عملیات‌هایی اجازه دسترسی دارند. منابع می‌تواند فایل‌ها، دیتابیس‌ها، جداول و سطرهای دیتابیس و... باشد. عملیات شامل اجرای یک تراکنش(خرید یک محصول)، انتقال پول از یک حساب به حساب دیگریا افزایش نرخ سود یک مشتری است.

Auditing[6] 3-1-1-1: یک نظارت و لاگ گیری موثر بهترین راه برای مقابله با انکارِ-نفوذ[۷] است. به عمل انکار انجام یک کار یا آغاز یک تراکنش از طرف User در اصطلاح Non-repudiation گفته می‌شود.

---

[1] Security is a path not a destination.
[2] احراز هویت
[3] End user
[4] مدیران
[5] تعیین مسئولیت
[6] زیر نظر گرفتن و ثبت رد پا
[7] Non-repudiation

بطور مثال: در یک سیستم فروش الکترونیک مکانیزم‌های-ضد-انکار[1] لازم است تا مشتری نتواند سفارش ۱۰۰ نسخه از یک کتاب خاص را انکار کند.

نمونه سوال: مکانیزم‌های ضد انکار چیست؟با مثال توضیح دهید.

4-1-1-1 Confidentiality[2]: محرمانگی که به آن Privacy[3] گفته می‌شود فرآیندی است که تضمین می‌کند داده‌ها به صورت محرمانه و خصوصی باقی بماند و کاربران غیر مجاز و استراق سمع کنندگان که جریان ترافیک شبکه را زیر نظر دارند به داده‌ها دسترسی نداشته باشند.

Encryption[4]: از عمومی‌ترین روش‌ها برای تامین محرمانگی است.

Access Control List (ACL)[5]: راه دیگری برای تامین محرمانگی است.

5-1-1-1 Integration[6]: محافظت می‌کند که داده‌ها در برابر ویرایش‌های آگاهانه و یا تصادفی[7] در بین شبکه‌ها است، تأمین یکپارچگی و صحت داده‌ها به کمک تکنیک‌های هشینگ و کدهای "تأیید هویت پیغام" انجام می‌شود.

نمونه سوال: روش های تأمین یکپارچگی و صحت داده را بیان کنید.

6-1-1-1 Availability[8]: از نگاه امنیت Availability یعنی سیستم برای کاربران مجازو قانونی همیشه در دسترس باقی بماند. هدف بسیاری از حمله‌کنندگان از حملات [9] Denial Of Service(DOS)این است که برنامه یا زیرساخت اجرای یک برنامه را از کار بیاندازد. و یا آن را آنقدر مشغول نگهدارد که دیگر کاربران امکان دسترسی به برنامه را نداشته باشند.

نمونه سوال: در مورد حملات DOS توضیح دهید.

2-1 تعریف اصطلاحات:

1-2-1 Threats: یک تهدید، یک اتفاق بالقوه و خطرناک است که می‌تواند به یک موضوع صدمه وارد کند. هر اتفاق بدی که برای سیستم شما بیافتد یک تهدید به حساب می‌آید.

---

[1] Non-repudiation mechanisms
[2] محرمانگی
[3] حریم خصوصی
[4] رمزنگاری داده‌ها
[5] لیست کنترل دسترسی
[6] یکپارچه و درستی
[7] Malicious
[8] در دسترس بودن داده ها
[9] جلوگیری از سرویس

Vulnerabilities 2-2-1: آسیب پذیری یک نقطه ضعف است که می‌تواند یک تهدید را در پی داشته باشد که این آسیب پذیری ممکن است بخاطر طراحی ضعیف، تصمیمات اشتباه، و از کار افتادن یا نا امن بودن تکنیک‌های امنیتی بوجود آیند. برای مثال: اعتبارسنجی[1] ضعیف ورودی‌های یک فرم.

: یک آسیب پذیری در لایه‌ی Application به حساب می‌آید که می‌تواند حملات را در پی داشته باشد.

Attack[2] 3-2-1 : به عملی گفته می‌شود که از طریق آن از یک نقطه‌ی آسیب پذیر به سیستم نفوذ می‌شود و یا یک تهدید برای سیستم به وجود می‌آید.

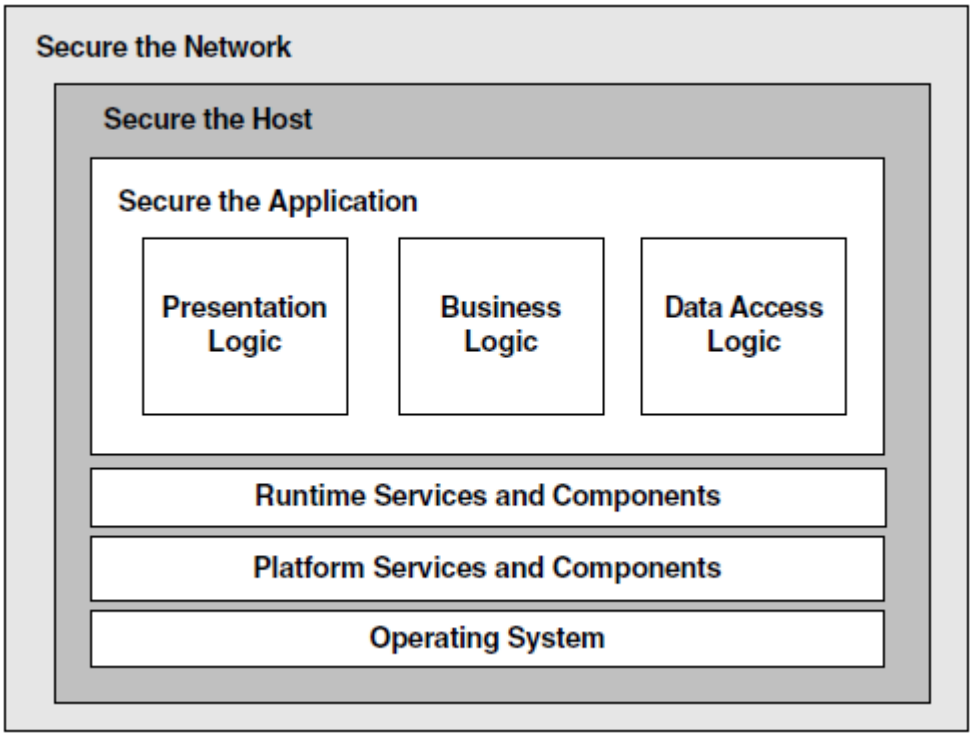
Ex[3]: بطور مثال ارسال یک ورودی مخرب در قالب یک فایل از طریق فرمی که روی یک سایت است یک نوع حمله به حساب می‌آید.

Threats-Model 4-2-1: شما نمی‌توانید یک برنامه‌ی تحت وبِ امن طراحی و ایجاد کنید مگر اینکه تهدیدهای خود را بدانید. یکی از فازهای مهم در ایجاد یک برنامه‌ی تحت وب، فاز مدل‌سازی تهدید است . تحلیل معماری برنامه شما و طراحی و مشخص کردن حوزه‌های آسیب پذیر بالقوه که ممکن است یک کاربر اشتباهاً و یا یک آتکِر از روی عمد امنیت سیستم شمارا به خطر بیاندازد.

[1] validation
[2] حمله
[3] نفوذ

۱- ۳ سه لایه **اصلی** امنیت: امنیت یک برنامه تحت وب در سه سطح یا لایه مورد بررسی قرار می‌گیرد:

- ✓ **در سطح شبکه**
- ✓ **در سطح هاست**
- ✓ **در سطح اپلیکیشن**



شکل ۱-۱ لایه‌های اصلی امنیت

فصل دوم

۲-۱ نکات مربوط به امنیت شبکه:

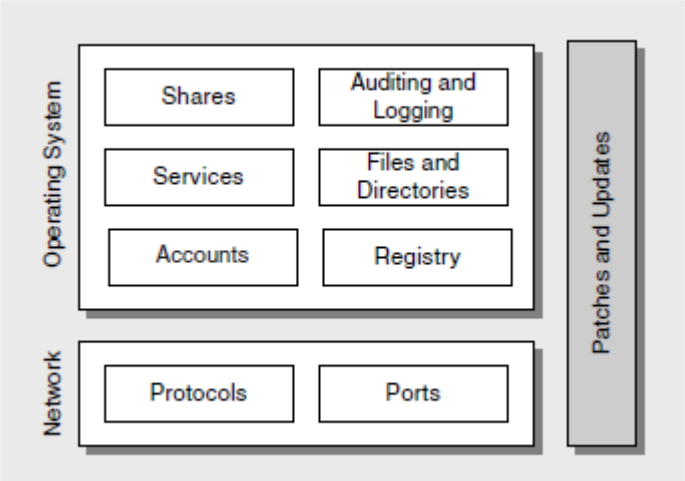
یک برنامه‌ی تحت وب امن بروی یک زیرساخت شبکه‌ی امن قرار می‌گیرد. زیرساخت شبکه شامل: روترها [i]، فایروال‌ها [ii] و سوئیچ‌ها [iii] است. موارد زیر مربوط به یک شبکه‌ی امن است:

- ✓ حملات مبتنی بر TCPIP
- ✓ واسطه‌های مدیریتی امن مانند مخفی بودن آدرس کنترل پنل سایت
- ✓ پسوردهای امن در بین مدیران سازمان

برای تامین امنیت شبکه باید با مفاهیمی مانند پورت‌ها، پروتکل‌ها و ارتباطات مخرب در شبکه آشنا باشید.

۲-۲ نکات مربوط به امنیت هاست:

امنیت هاست شامل امنیت وب سرور، اپلیکیشن سرور و دیتابیس سرور می‌شود. بطور مثال در مورد هاست‌های اجرا کننده کدهای PHP باید وب سرور که Apache است و اپلیکیشن سرور که PHP است و دیتابیس سرور که MYSQL است، هرسه امن باشد. گاهی نسخه‌های خالی از PHP یا Apache یا MYSQL دارای باگ‌هایی هستند که ممکن است کل برنامه‌های روی هاست را به خطر بیاندازد.



شکل ۲-۱ شاخه‌های مختلف پیکر بندی هاست

۲-۳ نکات مربوط به امنیت برنامه:

۲-۳-۱ فرآیند مدل سازی تهدید:

۲-۳-1-۱ گام اول: داشته‌ها را مشخص کنید.

۲-۳-۱-۲ گام دوم: یک نمای کلی از معماری سیستم مشخص کنید.

۲-۳-۱-۳ گام سوم: برنامه‌ای که نوشته‌اید را به بخش‌های مختلف تجزیه کنید.

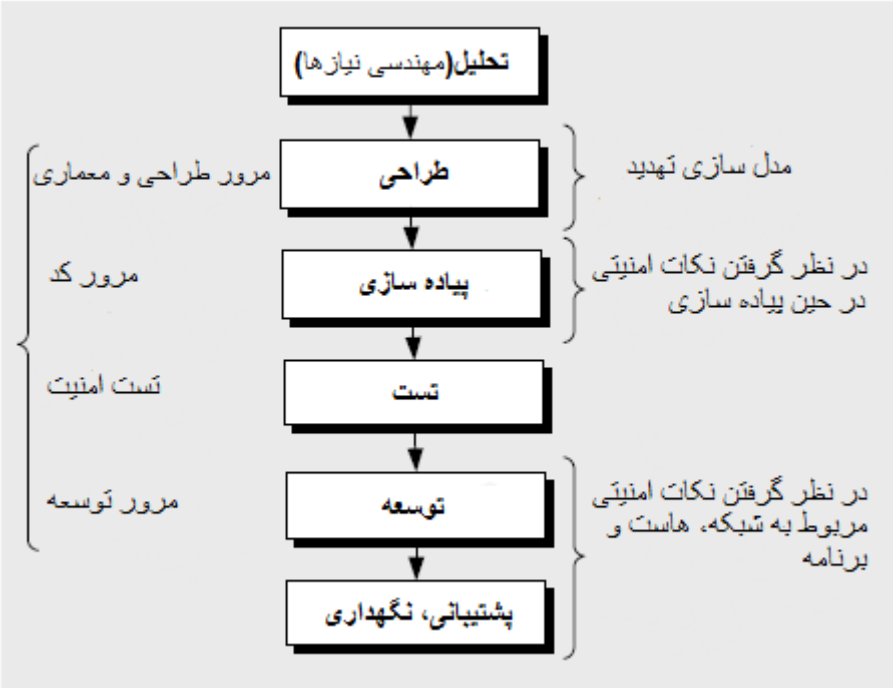
۲-۳-۱-۴ گام چهارم: تهدیدهای هربخش را مشخص کنید.

۲-۳-۱-۵ گام پنجم: تهدیدها را مستندسازی کنید.

۲-۳-۱-۶ گام ششم: هر تهدید را نسبت به اهمیت آن رتبه‌بندی کنید.

- ✓ **در گام اول**: داده‌هایی از سیستم که باید محافظت شوند را شناسایی و ارزش هر کدام را تعیین کنید.
- ✓ **در گام دوم**: از دیاگرام‌های ساده مانند مستطیل، فلش، جدول و ... برای مستندسازی معماری برنامه و زیر سیستم‌های آن و مرزهای امن و جریان داده استفاده کنید.
- ✓ **در گام سوم**: هرآنچه که به امنیت شبکه شما مرتبط است را مد نظر قرار دهید.مانند: زیرساخت‌های شبکه و هاست شما.
- ✓ **در گام چهارم**: همیشه اهداف یک حمله کننده[1] را در ذهن خود داشته باشید این کمک میکنه که در حین طراحی سیستم نقاط آسیب پذیر و تهدیدها را بهتر شناسایی کنید. یعنی بهتر است در حین طراحی سیستم امنیت آن را نیز تامین کنیم و نه ایمکه پس از طراحی سیستم به فکر نقاط آسیب‌پذیر باشیم. در حین کدنویسی میتوانید با کامنت‌هایی[2] که در بخش‌های آسیب‌پذیر یادداشت می‌کنید،کار روی آن بخش را به مراجعات بعدی واگذار کنید.
- ✓ **در گام پنجم**: هر تهدید را نسبت به اهمیت آن با غالب خاصی مستند سازی کنید. بطور مثال یک تهدید نیاز به تحقیقات گسترده و مستندات پیچیده‌تری دارد در حالی که برخی تهدیدات با حداقل توضیحات مستند سازی می‌شود.
- ✓ **در گام ششم**: برخی تهدیدات ممکن است بسیار حیاتی و فوری باشد و ریسک بالایی برای پروژه ایجاد کند. این نوع تهدیدات باید در اولین خروجی یا آپدیت برنامه رفع شود و برخی دیگر که ریسک بالایی ندارد می‌تواند به آپدیت بعدی مُحوِل شود.

[1] Attacker
[2] comment



شکل ۲-۲ رابطه‌ی امنیت با چرخه زندگی محصول

۲-۳-۲ چه کسانی در قبال امنیت نرم‌افزار مسئول هستند؟

طراحی و ساخت برنامه‌های امن یک کار دسته جمعی است و نقش‌های مختلفی در آن درگیرند. افراد درگیر در امنیت برنامه بطور مخفف RACI نامیده می‌شود که عبارتنداز:

Responsible[1]: افرادی که مسئول اجرا و پیاده‌سازی برنامه هستند.

Accountable[2]**:** افرادی که یک سری مسئولیت‌های کلی نسبت به تامین امنیت برنامه دارند.

Consulted: افرادی که ورودی‌های سیستم را تامین می‌کنند.

Keep Information: افرادی که مسئول اطلاع‌رسانی نکات مربوط به امنیت سیستم هستند.

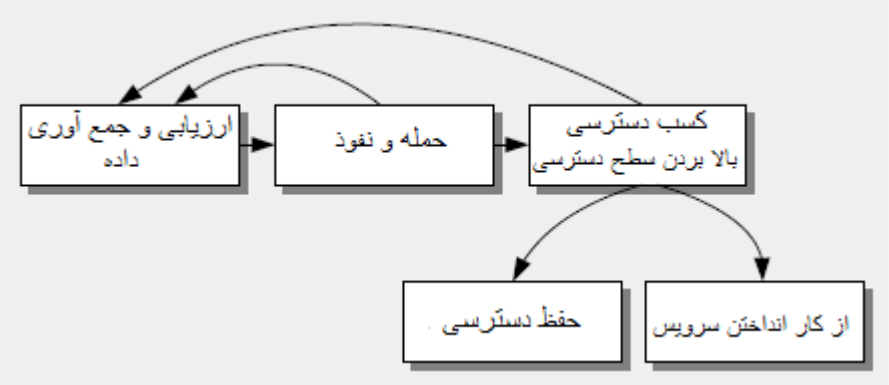
---

[1] مسئول
[2] پاسخگو

وظایف	معمار	مدیر سیستم	توسعه دهنده	تستر	امنیت تخصصی
سیاستهای امنیتی		R		I	A
مدل سازی تهدید	A		I	I	R
مفاهیم طراحی امنیت	A	I	I		C
معماری امنیت	A	C			R
مرور طراحی و معماری	R				A
پیاده سازی کد			A		R
تهدیدهای مربوط به تکنولوژی			A		R
مرور کد			R	I	A
تست امنیت	C		I	A	C
امنیت شبکه	C	R			A
امنیت هاست	C	A	I		R
امنیت برنامه	C	I	A		R
مرور توسعه	C	R	I	I	A

شکل ۲-۳ وظایف هریک از نقش‌های درگیر در یک برنامه

۲-۴ کالبد شکافی(تشریح) یک حمله:



شکل ۲-۴ تشریح یک حمله

۲-۴-۱ ارزیابی و جمع‌آوری داده‌ها[1]: اولین گامی که هر حمله کننده معمولا برمی‌دارد جمع‌آوری اهداف بالقوه و بدست آوردن ویژگی‌های آن هدف است. بطور مثال از اهداف بالقوه در یک برنامه، فرم‌های ثبت اطلاعات در سیستم است. منظور از ویژگی‌ها، پروتکل‌ها و سرویس‌هایی است که از آن نقطه‌ی نفوذ پشتیبانی می‌کند. هکر[2] از اطلاعات جمع‌آوری شده برای ایجاد یک پلان برای حمله استفاده می‌کند. بطور

[1] Survey and Assess
[2] نفوذگر

مثال یک نفوذگر که قصد دارد از طریق XSS حمله کند ابتدا بررسی می‌کند که آیا فرم‌های صفحه وب هر چیزی را به عنوان ورودی قبول می‌کند و آیا همان را دقیقا در خروجی چاپ می‌کند.

۲-۴-۲ حمله و نفوذ[1]: بعد از جمع‌آوری نقاط بالقوه گام بعد تست نفوذ از آن نقطه است. برای یک نفوذگر ساده‌ترین راه برای نفوذ همان بخش‌هاییست که کاربران قانونی برای ورود به سیستم استفاده می‌کنند. بطور مثال صفحه Log in.

۲-۴-۳ کسب دسترسی[2]: بعد از اینکه نفوذگر امکان نفوذ به برنامه یا شبکه را پیدا کرد، از طریق تزریق کد به برنامه و یا ایجاد یک سشن معتبر با سیستم عامل سطح دسترسی خود را به سرعت بالا می‌برد. بطور مثال یک اکانت ایجاد می‌کند و آن را عضو گروه Administrator می‌کند تا بالاترین سطح دسترسی را از آنِ خود کند. بنابراین برای شما به عنوان برنامه نویسِ برنامه‌های تحت وب در نظر گرفتن حداقل دسترسی‌ها خیلی مهم است.(به هر کسی همان قدر دسترسی دهید که نیاز دارد)

۲-۴-۴ حفظ دسترسی[3]: به محض اینکه نفوذگر به سیستم دسترسی پیدا کرد روالی را طی می‌کند تا دسترسی‌های آینده را آسان‌تر و ردپای خودش را مخفی کند. برخی روش‌های معمول برای این کار داشتن برنامه‌های درِ پشتی و یا استفاده از یک اکانتی که حفاظت قوی روی آن وجود ندارد و حساسیت ایجاد نمی‌کند. ممکن است برای پوشش ردپا نیاز باشد لاگ‌های سیستم پاک شود و یا ابزارهایی که تزریق شده است مخفی می‌شود. بنابراین در طراحی یک سیستم امن، باید لاگ فایل‌ها امن نگهداشته شوند و هربار بطور منظم آنالیز شوند. آنالیز لاگ‌های فایل بسیاری از نشانه‌های تلاش برای تخریب و نفوذ را آشکار می‌کند. مانند ابزار Event viewer و Laster visitors در ویندوز (آخرین بازدید کنندگان سایت را نمایش می‌دهد) و ابزار Webalizer وAnalog stats (جزئی‌ترین دسترسی‌ها به فایل‌ها و پوشه‌های سایت را نمایش می‌دهد.)

تحقیق در مورد شل c99 تحقیق کنید یک کنترل پنل(هاست) پیدا کنید و برنامه‌های ثبت لاگ را در آن بررسی کنید.

۲-۴-۵ از کار انداختن سرویس[4]: حمله کنندگانی که نتوانند یک دسترسی قوی به سایت پیدا کنند اغلب به عنوان تیر آخر، حملات DOS[5] (خود داری از خدمات) را شکل می‌دهند. حملاتی مانند SYN

[1] Exploit and Penetrate
[2] Escalate Privileges
[3] Maintain Access
[4] Deny Service
[5] Denial of service

flood که در آن حمله کننده یک خَروار تقاضای TCP SYN به سرور یا برنامه ارسال می‌کند. این کار باعث می‌شود کاربرانِ دیگر نتوانند با شبکه‌ی سرور یک کانکشن برقرار کند.

۲-۵ دسته بندی تهدیدات:

با توجه به اینکه حمله‌ها و تکنیک‌های حمله‌ها بسیار متنوع هستند، بهتر است بجای بررسی تک تکِ حملات، اهداف حمله کنندگان را مورد توجه قرار دهیم.

۲-۵-۱ دسته‌بندی حملات از نظر اهداف حمله کننده:

STRIDE: شرکت مایکروسافت اهداف مختلف حمله کنندگان را در کلمه STRIDE خلاصه کرده است. این کلمه مخفف کلمات زیر است:

2-1-5-1 Spoofing[1]: (حقه ، کلک ، خود را جای کسی جازدن) تلاش برای کسب دسترسی به یک سیستم از طریق یک هویت جعلی است.. این کار می‌تواند از طریق دزدیدن Credential[iv] کاربر یا تغییر IP یا MAC address[v] به IPیا MAC یک کاربرِ مجاز صورت گیرد.

2-1-5-2 Tampering[2]: (دستکاری) به ویرایش غیر مجاز داده‌ها گفته می‌شود. بطور مثال وقتی داده‌ای بین دو کامپیوتر در شبکه در حال جریان است، در بین راه آن را دستکاری می‌کند.

2-1-5-3 Repudiation: (انکار) به قدرت کاربر چه کاربر قانونی و مجاز چه غیر قانونی برای انکار عملیات یا تراکنش خاصی که انجام داده Repudiation گفته می‌شود بدون وجود سرویس‌های ردگیری، اثبات حمله مشکل است.

2-1-5-4 Information Disclosure[3]: به انتشار ناخواسته‌ی اطلاعات خصوصی گفته می‌شود. برای مثال یک کاربر اطلاعات فایل یا جداولی که به او دسترسی داده نشده را می‌بیند و یا می‌تواند داده‌های ردٌ و بدل شده در شبکه را مانیتور کند. برخی نمونه‌های این نوع حمله عبارتند از استفاده از فیلدهای مخفی در فرم‌ها و یا کامنت‌هایی که در یک صفحه‌ی وب قرار داده شده که در حقیقت یک اتصال به دیتابیس است و یا Exception handlingهای ضعیف که ممکن است منجر به افشای جزئیات داخلی سیستم شود.

2-1-5-5 Denial of service[4]: انکار خدمات یا DOS فرآیندی است که یک سیستم یا برنامه را از دسترس خارج می‌کند. برای مثال این کار می‌تواند از طریق بمب باران یک سرور با

---

[1] *Spoofing* is attempting to gain access to a system by using a false identity.
[2] *Tampering* is the unauthorized modification of data.
[3] *Information disclosure* is the unwanted exposure of private data.
[4] *Denial of service* is the process of making a system or application unavailable.

تقاضاهایی که منابع سیستم را برای طولانی مدت درگیر می‌کند و یا ارسال داده‌های مضر که باعث کِرَش کردن فرآیندهای برنامه می‌شود. مثال دیگر: یک کاربر اقدام به دانلود مکرر فایل‌های حجیم از روی یک سایت کند و یا آن فایل را در محیط‌های عمومی ب‌رای دانلود بگذارد تا حدی که پهنای باند ماهانه سایت به اتمام برسد.

6-1-5-2 Elevation of privilege[1]: (بالا بردن سطح دسترسی) این حمله زمانی اتفاق می‌افتد که یک کاربر با دسترسی محدود عامل احراز هویت یک کاربر قوی‌تر را به دست آورده و از آن طریق به برنامه دسترسی پیدا کند.

۲-۵-۲ اقدامات متقابل در برابر تهدیدات STRIDE :

1-2-5-2 اقدامات متقابل در برابر Spoofing user identify:

- از احراز هویت‌های قوی استفاده کنید. بطور مثال احرازهویت از طریق کارت هوشمند علاوه بر پین کد در بحث Encryption در ویندوز
- عوامل رمزگونه مانند پسوردِ کاربران را در قالب متن‌ساده[2] ذخیره نکنید و همچنین کاربران را از انتخاب پسووردهای ساده منع کنید.
- Credentialها را به حروف متن ساده روی شبکه ردّو بدل نکنید.
- ترجیحا کوکی‌ها یا داده‌های دیگر را با SSL محافظت کنید.

2-2-5-2 اقدامات متقابل در برابر Tampering with data:

- داده ها را به صورت هش شده ویا به صورت امضا شده و یا Obfuscate[3] شده ذخیره کنید.
- از امضاهای دیجیتال قبل از دستکاری داده‌ها استفاده کنید.
- سطح دسترسی داده‌هایی که باید ویرایش شود را تا حد ممکن بالا در نظر بگیرید.
- لینک‌های ارتباطی را با پروتکل‌های ضد دستکاری امن کنید.
- از پروتکل‌های یکپارچه‌سازی پیغام استفاده کنید.

[1] *Elevation of privilege* occurs when a user with limited privileges assumes the identity of a privileged user to gain privileged access to an application.
[2] Plain text
[3] درهم

3-2-5-2 اقدامات متقابل در برابر Repudiation:

- از روش‌های ثبت ردپای امن استفاده کنید. بطور مثال ثبت IP و مشخصات مرورگر و سیستم‌عامل کاربر و مسیرهایی که در حین استفاده از سیستم مشاهده کرده نمونه‌هایی از تکنیک‌های ثبت ردپا هستند.
- برای جلوگیری از تغییراتی که پس از آن امکان انکار تغییرات وجود دارد از امضای دیجیتال استفاده کنید.

4-2-5-2 اقدامات متقابل در برابر اِفشای اطلاعات[1]:

- از روش‌های احرازهویت قوی استفاده کنید.
- از روش‌های رمزنگاری قوی استفاده کنید.
- از پروتکل‌های محرمانگی[2] پیغام برای امن کردن لینک‌های ارتباطی استفاده کنید.
- موارد رمزگونه مانند رمزهای عبور را بصورت متن ساده ذخیره نکنید.

روش اول: روش‌های ناخوانا کردن اطلاعات یا Data Camper: در این حالت که معمولا در مورد زبان‌های جاوا اسکریپت و CSS و HTML که به سیستم کاربران منتقل می‌شود اتفاق می‌افتد. سند سفید کدها حذف می‌شود تا هم خوانایی خود را از دست بدهد و هم فایل حاوی کدها حجم کم‌تری اشغال کند. برای مثال سایت www.jscompress.com کدهای جاوااسکریپت را فشرده می‌کند. البته سایت‌هایی مثل www.jsbutifier.org نیز هستند که برعکس سایت قبلی عمل می‌کند. بنابراین این روش-روشِ منطقی برای حفاظت از کدها نیست. اما اگر مخاطبان شما کاربران حرفه‌ای نیستند می‌تواند یک مانع به حساب بیاید. در مورد زبان‌های HTMLو CSS نیز سایت‌های مشابهی وجود دارد که با جستوجو در اینترنت بصورت مثلا: PHP minifier و PHP butifier وجود دارد.

- **روش دوم:** obfuscation: در این روش کدهای برنامه‌نویسی با یک سری کدهای جدید و درهم جایگزین می‌شود و یک تابع برای تفسیر آن به کدها اضافه می‌شود. این روش نیز بطور ۱۰۰٪ جلوگیری از ویرایش کدهارا تضمین نمی‌کند.
- **روش سوم:** در این روش نام متغیرها و ثوابتی که در پروژه بکار رفته را با یکسری نام‌های بی معنی جایگزین می‌کند تا درک آن‌ها برای حمله کننده آسان نباشد.

---

[1] Information disclosure
[2] Confidentiality

- **روش چهارم:** استفاده از روش‌های Encrypt کردن کدها: در این روش کدها بطور کامل رمزنگاری می‌شود اما مشکل بزرگ این است که سرور سایت باید حتما به یک Decryptor مانند Zend مجهز باشد، که معمولا اینطور نیست.

5-2-5-2 اقدامات متقابل در برابر Denial Of Service:

- از تکنیک‌های کنترل پهنای باند و منابع استفاده کنید تا فرآیندهای مختلف، بیش از حدِ مجاز وارد پهنای باند نشوند.
- ورودی‌های برنامه را اعتبار سنجی و آن‌ها که نامعتبر هستند را فیلتر کنید.

6-2-5-2 اقدامات متقابل در برابر بالا رفتن سطح دسترسی حمله کننده[1]:

- حداقل دسترسی‌ها را به سرویس‌ها و کاربران جهت دسترسی به منابع و فرآیندها بدهید. (فراموش نکنید که به هر کس همانقدر دسترسی بدهید که نیاز دارد.)

[1] Elevation of privilege

فصل سوم

۳-۱ تهدیدات مربوط به شبکه و اقدامات متقابل آن:

اجزای اصلی زیرساخت شبکه شما عبارتنداز فایروال‌ها، روترها و سوئیچ‌ها. یک هکر ممکن است از طریق پیکربندی ضعیف دستگاه‌های شبکه نفوذ کند. بطور مثال ممکن است مدیر شبکه تنظیمات پیشفرض دستگاه‌ها را تغییر نداده باشد. (برای مثال ممکن است نام کاربری و رمز عبور ورود به کنترل پنل دستگاه همان مقادیر پیش‌فرض که معمولا مقادیر Admin و Admin هستند قرار داده شده باشند یعنی تغییر داده نشده باشد.)

عدم نصب آخرینPatch های امنیتی روی دستگاه‌ها می‌تواند مشکل‌زا باشد.

۳-۲ مهم‌ترین تهدیدات در سطح شبکه:

- Information gathering[1]
- Sniffing[2]
- Spoofing[3]
- Session hijacking[4]
- Denial of service[5]

3-2-1 Information gathering[6]:

دستگاه‌های شبکه می‌توانند از طریق برخی نرم‌افزارها یا برخی دستورات در شبکه کشف شوند. هکرها معمولا از طریق اسکن-پورت‌ها[7] کار را شروع می‌کنند پس از اینکه پورت‌های باز را شناسایی کردند از طریق عملیاتی که به آن Banner grabbing می‌گویند. نوع دستگاه و سیستم عامل و ورژن برنامه‌ای آن را شناسایی می‌کنیم پس از جمع‌آوری این اطلاعات هکر شروع به نفوذ از طریق ضعف‌های امنیتی که در آن دستگاه‌ها یا سیستم‌عامل‌های می‌شناسند،

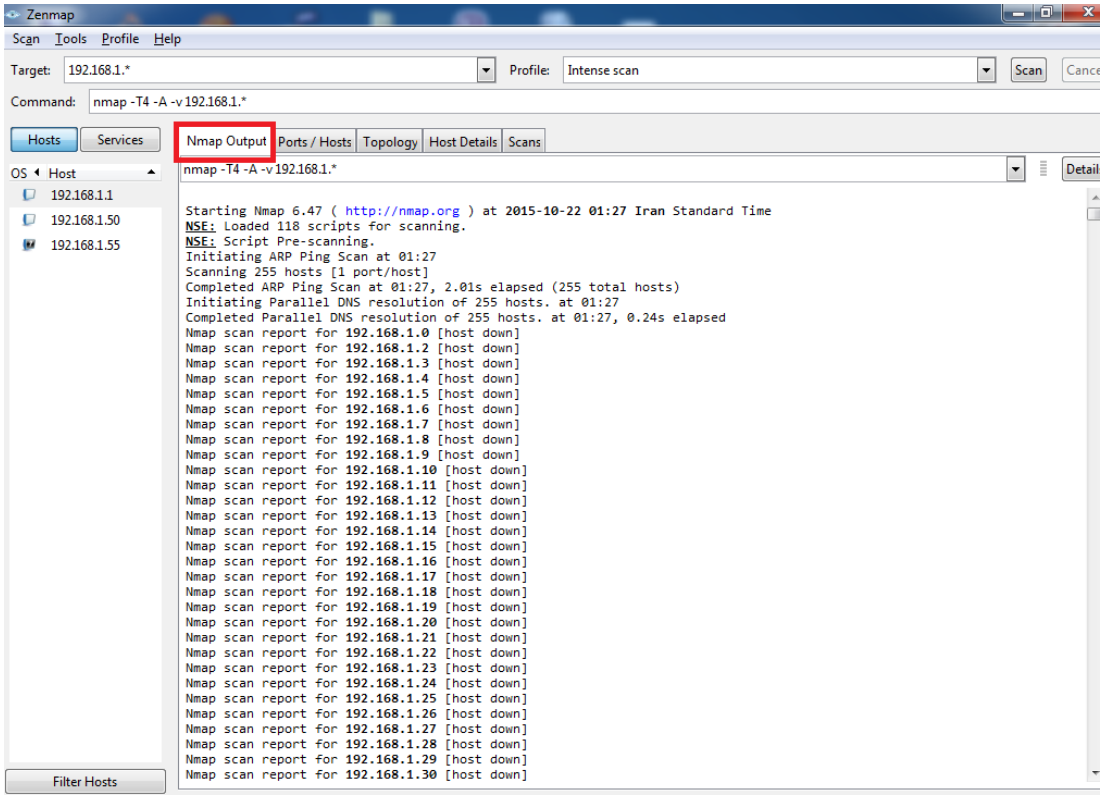
---

[1] جمع آوری اطلاعات
[2] شنود
[3] حُقه
[4] دزدی جلسه
[5] خود داری از خدمات
[6] جمع آوری اطلاعات
[7] Port scanning

می‌کنند. (برنامه‌های Advanced IP scanner و Advanced Port Scanner و Nmapبرای بدست آوردن پورت و آی‌پی می‌باشد.)

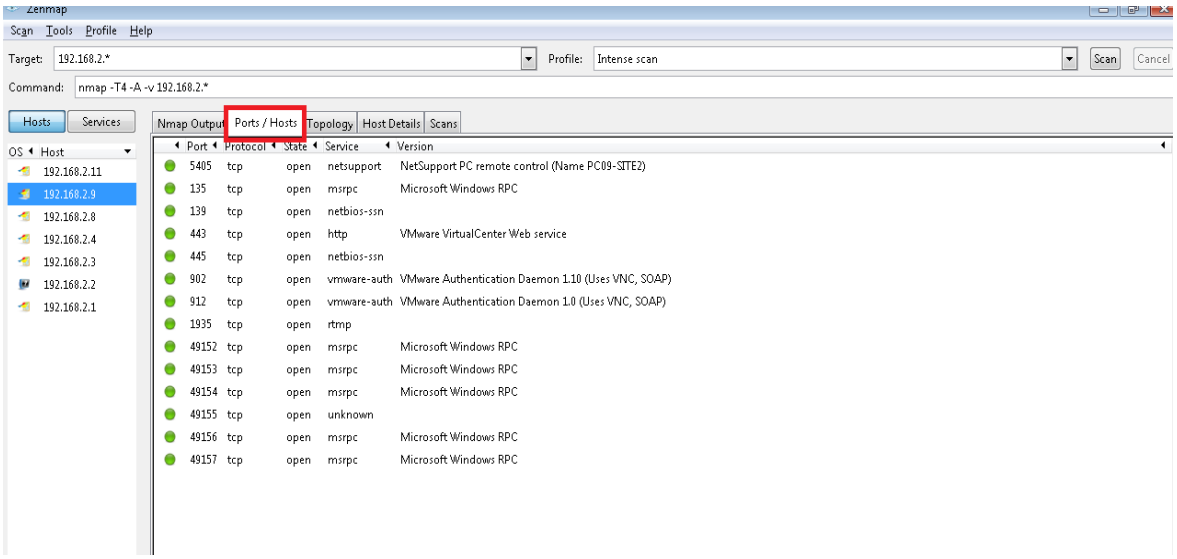
۳-۲-۱-۱ کار با برنامه Nmap:

- Nmap Output: در این مرحله در حال اسکن هاست‌ها می‌باشد.



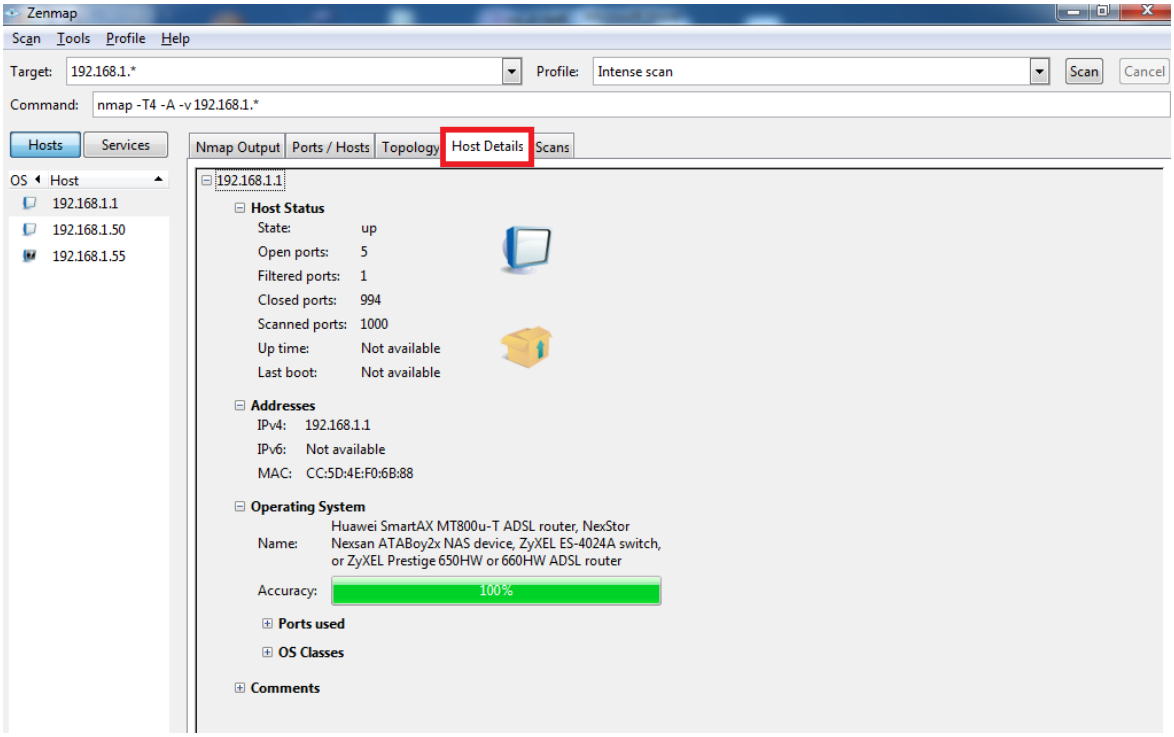
شکل ۳-۱ اسکن هاست‌ها

- Ports / Hosts:



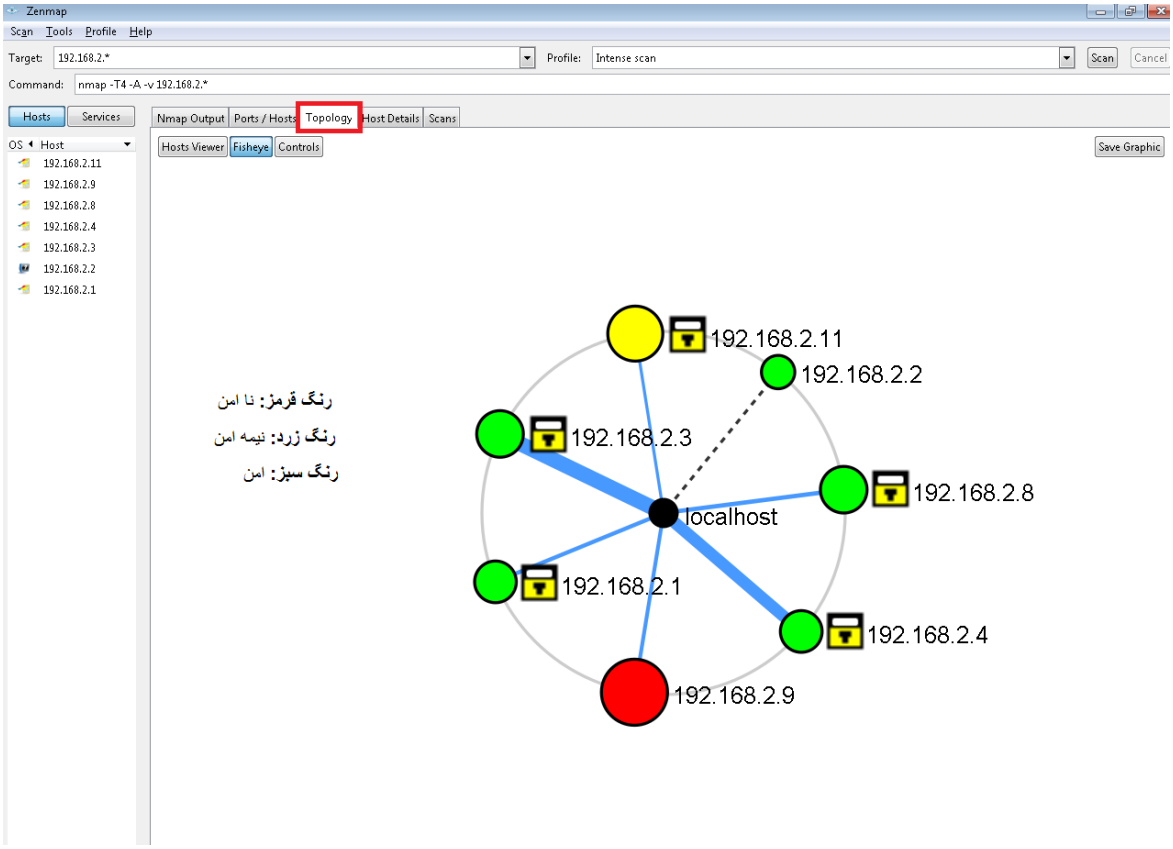
شکل ۳-۲ مشخصات هاست‌ها و پورت‌های موجود

- Host Details: در تصویر زیر جزئی‌ترین مشخصات یک هاست با کمک برنامه Nmap بدست آمده است.



شکل ۳-۳ جزئی‌ترین مشخصات هاست مورد نظر

- Topology: بطور مثال در تصویر زیر می‌توان توپولوژی کلی شبکه را با کمک برنامه Nmap فهمید.



شکل ۳-۴ توپولوژی کلی شبکه

۳-۲-۱-۲ آشنایی با فایروال:

فایروال یا دیوار آتش یک دیوارِ مجازیِ امنیتی بین یک کامپیوتر و شبکه‌های متصل به آن است. این دیوار $۲^{۱۶}$ یعنی ۶۵۵۳۶ پورت یا درگاه ورود و خروج دارد. هر برنامه‌ای که بخواهد از طریق شبکه با یک سیستم یا برنامه‌ی دیگر در ارتباط باشد باید حتما از طریق یکی از این درگاه‌ها تبادل اطلاعات کند یعنی باید یکی از این درگاه‌ها را برای خود رزرو کند. هرگز نمی‌توان دو برنامه تصور کرد که بطور همزمان از یک پورت استفاده کنند. برنامه‌ها و سرویس‌های مختلف مانند مرورگرها برنامه‌های ارسال و دریافت ایمیل، برنامه‌های چت و... برای اینکه بتوان از طریق شبکه با سرور خود در ارتباط باشند و سیستم‌عامل بفهمد که پکتی که اکنون وارد سیستم شده مربوط به کدام برنامه باز است باید حتما در هدِر TCP شماره پورت مبدأ را برابر با شماره مختص خودشان و شماره پورت مقصد را برابر با شماره‌ای که سرور آن‌ها از آن پورت، در اصطلاح شنود[1] می‌کند قرار دهد. بنابراین هر برنامه تحت شبکه باید شماره پورت یکتای مختص به خود را از موسسه‌ی IANA(Internet Assigned Number Authority)[2] تهیه و رِزرو کنند. لیست پورت‌ها و

[1] Listen
[2] مسئول اعداد نسبت داده شده‌ی اینترنتی

برنامه‌های مربوط به هرکدام در آدرس www.iana.org/assignment/port-number موجود است.
برخی از پورت‌ها، پورت‌های عمومی به حساب می‌آیند.

۲۱	FTP
۲۳	TELNET
۲۵	SMTP
۸۰	HTTP
۱۱۰	POP3
۴۴۳	HTTPS

شکل ۳-۵ مهم‌ترین پورت‌ها

3-1-2-3 Network profile:

از ویندوز ۷، مایکروسافت شبکه‌ها را به چهار دسته‌ی زیر تقسیم کرد:

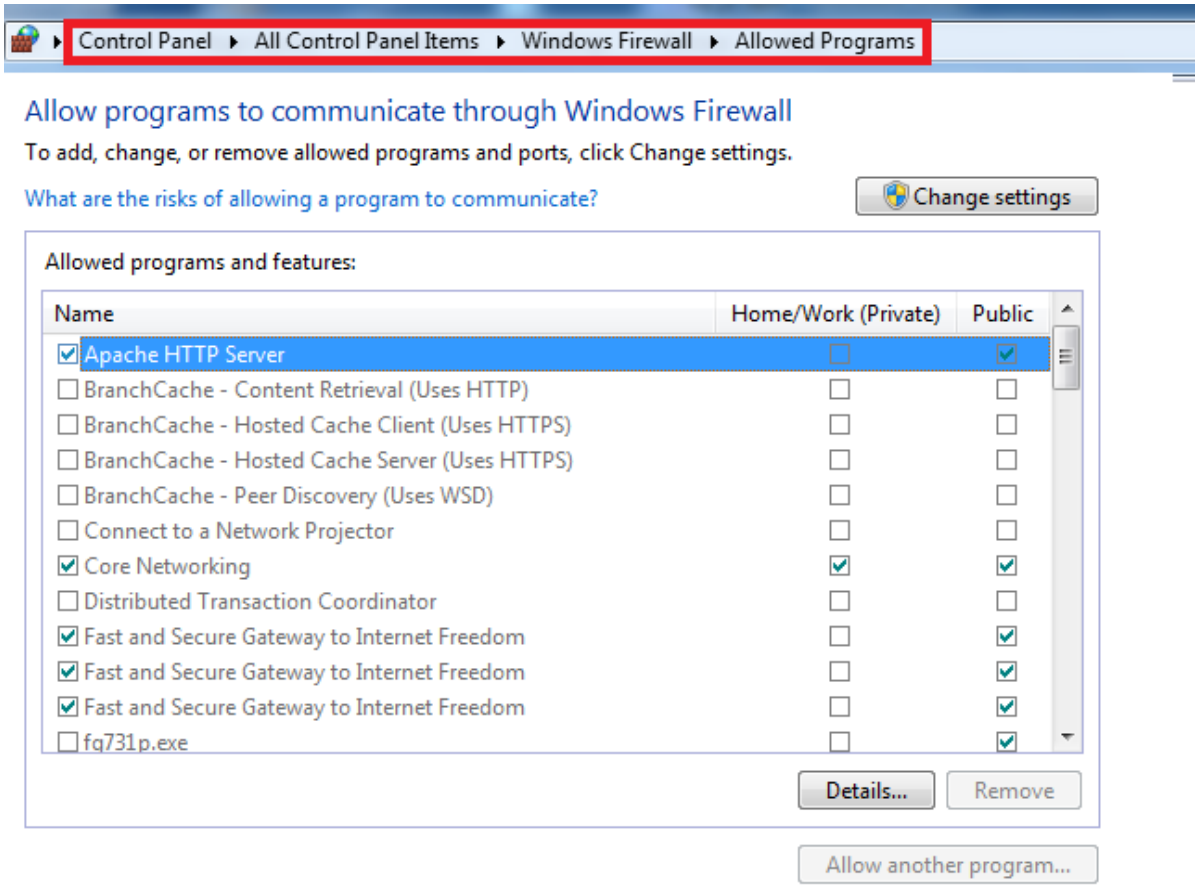
- Home[1]
- Work[2]
- Public[3]
- Domain[4]

تفاوت هر کدام از این پروفایل‌ها در تنظیمات فایروال است، طبیعتا هرچی از بالا به پایین حرکت کنیم فایروال سخت‌گیرانه‌تر عمل می‌کند. برای تغییر پروفایل در کنترل پنل در بخش Network and sharing center برای هریک از کارت‌های شبکه پروفایل مخصوص به آن را انتخاب کنید.

بستن یا باز کردن پورت‌های برنامه در فایروال به دو روش انجام می‌شود:

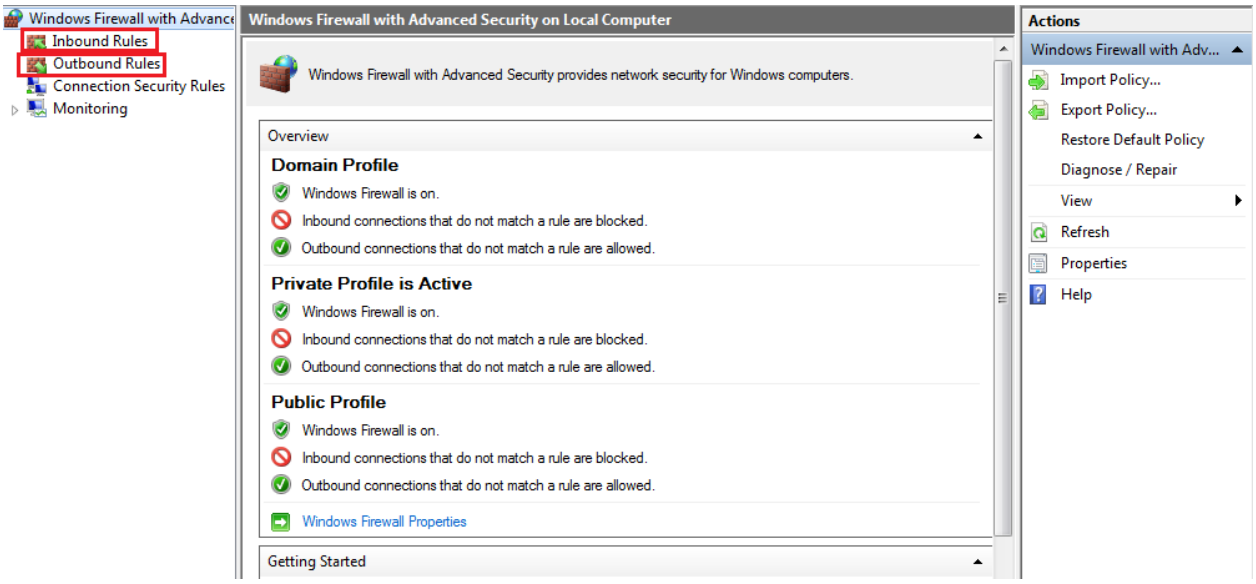
- روش اول: طبق عکس زیر برای اجازه دسترسی به پورت باید آن را تیک بزنیم.

[1] مخصوص شبکه‌های خانگی
[2] مخصوص شبکه‌های کاری
[3] مخصوص شبکه‌های عمومی(کافی نت)
[4] مخصوص شبکه‌های کلاینت/سروری



شکل ۳-۶ روش بستن یا باز کردن پورت‌های برنامه

- روش دوم: در فایروال باید وارد گزینهAdvanced setting شویم. این روش پیشرفته‌تر از روش قبل است.



شکل ۳-۷ روش بستن یا باز کردن پورت‌های برنامه

در این روش باید یک Rule[1] تعریف شود: دو نوع Rule در فایروال وجود دارد:

- Inbound Rule[2] : شامل قوانینی است که از ورود داده‌های ناخواسته به سیستم جلوگیری می‌کند.
- outbound Rule[3] : شامل قوانینی است که از خروج داده‌های ناخواسته از سیستم جلوگیری می‌کند.

۳-۲-۲ راه‌های متقابل در برابر تهدیدات Information Gathering[4] :

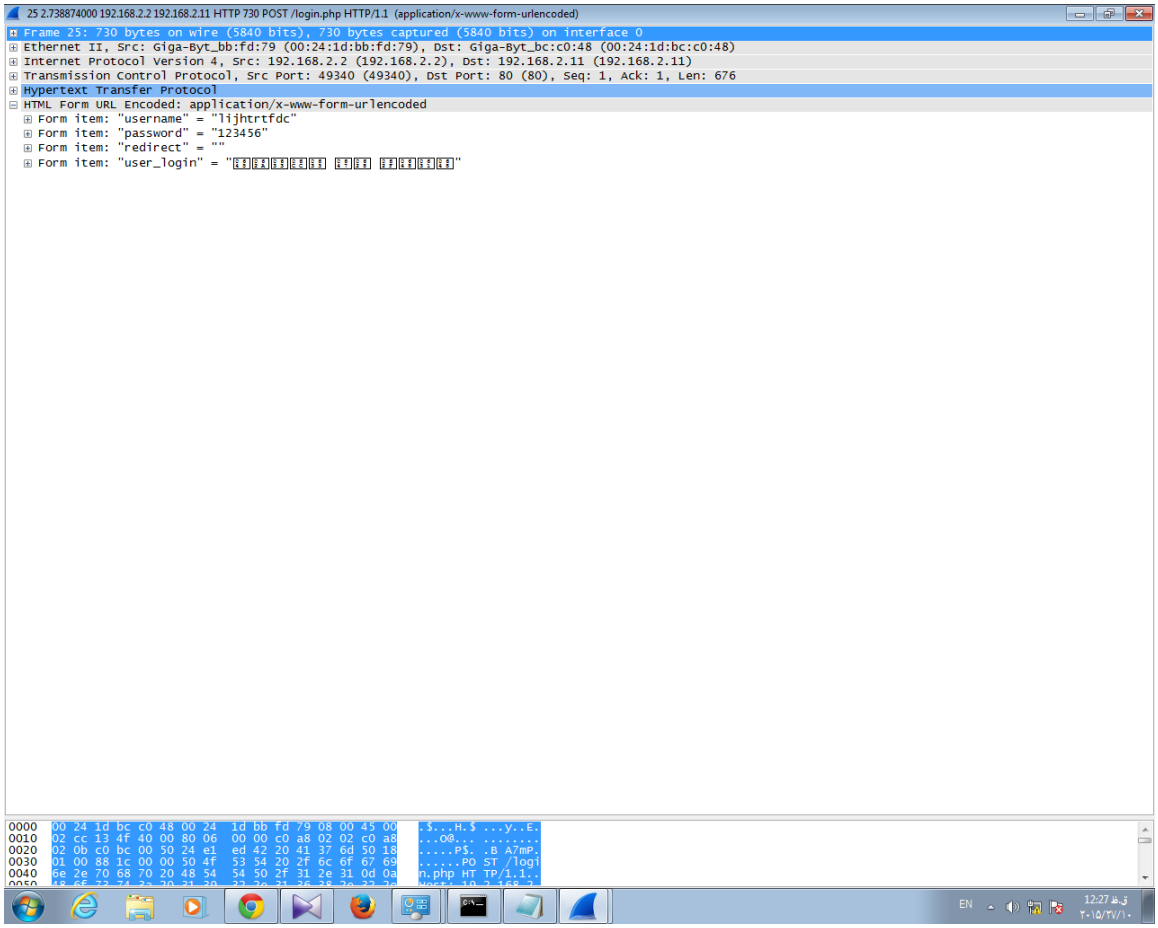
- روترها و دیگر دستگاه‌هایی که کاربرانِ مشخصی باید به آن‌ها متصل شوند را طوری تنظیم کنید، که فقط همان کاربران بتوانند به دستگاه متصل شوند. (بطور مثال از امکان Mac filtering استفاده کنید هم‌چنین می‌توان برخی دستگاه‌ها را در شبکه برای افراد غیر مجاز invisible کرد.)
- سیستم‌عامل‌هایی که نرم‌افزارهای شبکه را میزبانی می‌کنند را به درستی پیکربندی کنید. (بطور مثال فایروال سیستم‌عامل را طوری تنظیم کنید که، پروتکل‌هایی که هرگز استفاده نمی‌شوند و پورت‌هایی که لزومی ندارد فعال باشد، بسته باشند.)

۳-۲-۳ Sniffing[5]: به عملیات زیر نظر گرفتن ترافیک شبکه برای بدست آوردن داده‌هایی مانند پسوردهای تایپ شده یا اطلاعات مربوط به پیکربندی شنود یا استراق سمع[۶] گفته می‌شود.حتی نفوذگر می‌تواند پکت‌هایی که با الگوریتم‌های هش سبک رمزگذاری[۷] شده‌اند را رمزگشایی کنند در حالی که شما فکر می‌کنید داده امن است . برای سرقت پکت‌ها، سارق باید حتما در مسیر بین کلاینت و سرور قرار گیرد.

به نرم‌افزارهایی که برای این کار استفاده می‌شود در اصطلاح Packet sniffer گفته می‌شود که Wire shark یکی از آن‌ها به حساب می‌آید. بطور مثال می‌شود در تصویر زیر مشاهده می‌کنید که کاربر چه صفحه‌ای را درخواست کرده و چه یوزرنیم یا پسوردی را وارد کرده است:

---

[1] قانون
[2] قوانین درون مرزی
[3] قوانین برون مرزی
[4] جمع آوری اطلاعات
[5] شنود
[6] Eaves dropping
[7] Decipher or decrypt



شکل ۸-۳ مشاهده یوزرنیم و پسورد کلربربا استفاده از wire shark

سوال: یک وب‌سرور روی سیستم خود ایجاد کنید، از کاربر دیگری در شبکه بخواهید به وب‌سرور سیستم شما متصل شده و فرمی که شما روی وب‌سرور تعبیه کردید را پر کند. با استفاده از Wire shark داده‌هایی که کاربر در فرم وارد کرده را شنود کنید.

به عنوان تحقیق: در مورد a-man-in-the-middle تحقیق کنید.

تحقیق: نرم افزار packet sniffer برای Https پیدا کنید.

۳-۲-۴ اقدامات متقابل علیه **Sniffing**:

- امنیت فیزیکی سازمان و بخش‌بندی[1] شبکه را قوی کنید این اولین گام حفاظت از ترافیک شبکه در برابر استراق سمع‌های درون سازمان است.

[1] Network Segmentation

- ارتباطات را بطور کامل رمزنگاری کنید، بطور مثال یوزرنیم و پسوردهایی که بین کلاینت‌ها و سرور ردو بدل می‌شوند از طریق پروتکل‌هایی مانند SSL ویا IP Sec[1] [vi] رمزنگاری شود. این کمک می‌کند که پکت‌ها برای سارق غیرقابل استفاده باشد.

5-2-3 Spoofing[2]:

راه دیگر جمع‌آوری اطلاعات در شبکه ابزاری برای مخفی کردن هویت حقیقی یک نفوذ در شبکه است، برای این نوع حمله نفوذگر از یک آدرس جعلی[3] که آدرس حقیقی را نمایان نمی‌کند، استفاده می‌کند. بطور مثال برای اینکه از امکانات شبکه‌ی محلی در لپ‌تاپ خود استفاده کنند Mac address یا IP لپ‌تاپ را برابر با Mac address یا IP یکی از افراد سازمان می‌کند. در نتیجه تمام اقدامات او به نام آن طعمه ثبت خواهد شد.

۶-۲-۳ اقدامات متقابل علیه Spoofing:

- با استفاده از نرم‌افزارهای Proxy که از ورود پکت‌هایی که از IPهای داخل سازمان از جاهایی که انتظار آن را دارید آن پکت از IPهای خارجی جلوگیری کنیم. بطور مثال می‌توان در یک وب سرور از ورود IPهای Private جلوگیری کرد تا افراد فقط از طریق اینترنت یعنی با IP عمومی بتواند وارد شود.
- از خروج پکت‌هایی که بنظر می‌رسد از یک IP داخلی نا معتبر و غیر مجاز نشأت گرفته‌اند جلوگیری کنیم.

آشنایی با فایل htaccess در وب سرورها:

یک فایل پیکربندی[4] مخصوص وب‌سرویس‌هایی است که از Apache استفاده می‌کنند، این فایل در یک پوشه قرار می‌گیرد و تنظیماتی روی آن پوشه و زیرپوشه‌های آن انجام می‌دهد.

برخی از کارهایی که می‌توان با htaccess انجام داد:

- **مدیریت Error documents:**

می‌توانیم کاربر را در صورت مواجه شدن با ارورهای وب‌سروری به صفحات مدنظر خودمان هدایت کنیم.

---

[1] Internet Protocol Security
[2] حُقه
[3] Fake
[4] Configuration

برخی از ارورهایی که ممکن است به واسطه‌ی وب‌سرور مشاهده شود عبارتند از:

ارور ۴۰۱: به معنی Unauthorized[1]، در صورتی رخ می‌دهد که کاربر نام کاربری و پسورد ورودی پوشه یا فایل را درست وارد نکرده باشد.

ارور ۴۰۳: به معنی Forbidden، هنگامی رخ می‌دهد که کاربر امکان مشاهده‌ی مستقیم محتوای یک پوشه را نداشته باشد.

ارور ۴۰۴: به معنی Not found زمانی رخ می‌دهد که فایل یا پوشه (مسیر) فراخوانی شده وجود نداشته باشد.

ارور ۵۰۰: به معنی Internet server error زمانی رخ می‌دهد که تنظیمات وب‌سرور مناسب نباشد[2].

با استفاده از قطعه کد زیر در فایل htaccess می‌توان ارورهای بالا را به صفحات دلخواه ارجاع داد:

```
ErrorDocument 401/error_pages/401.html
ErrorDocument 404/error_pages/404.html
ErrorDocument 500/error_pages/500.html
```

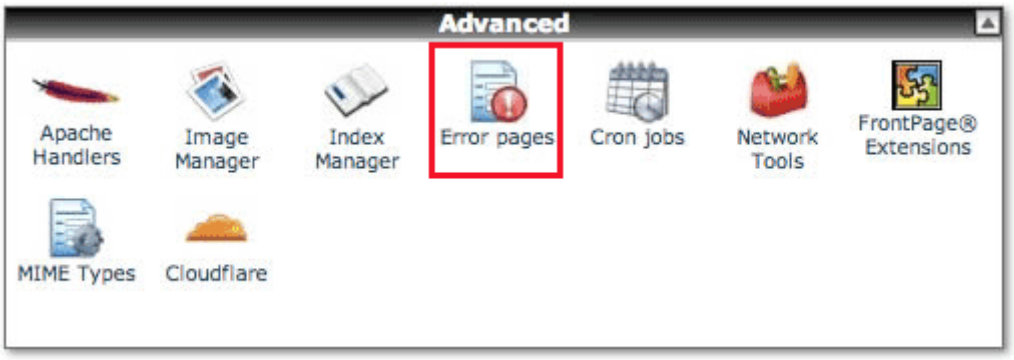
ارور ۴۰۳: یکی از راه‌های جم‌آوری اطلاعات، جلوگیری از لیست شدن محتویات یک پوشه است که می‌توان آن را از طریق فعال کردن Directory index در فایل htaccess و یا قرار دادن یک فایل index.html در آن پوشه تأمین کرد.نتیجه خاموش کردنDirectory index مشاهده ارور ۴۰۳ است.

ساختار کلی:

```
ErrorDocument کدارور/نظر مورد صفحه
```

در اکثر کنترل پنل‌ها ابزارهایی وجود دارد که کار با htaccess به صورت گرافیکی ممکن می‌کند. گزینه error pages در کنترل پنل Cpanel برای مدیریت صفحات ارور است.

[1] غیر مجاز
[2] Misconfiguration



جدول ۳-۹ گزینه error pages در کنترل پنل Cpanel

- **Redirect**: با استفاده از این امکان می‌توان تنظیم کرد که کاربر با فراخوانی یک مسیر به مسیر دیگری هدایت شود. این کار ممکن است بخاطر تغییر یک مسیر قدیمی که کاربران و یا موتورهای جستوجو قبلا با آن کار می‌کرده‌اند انجام شود. و یا کوتاه کردن یک آدرس طولانی و یا تغییر مسیر کاربر از مسیرهای مختلف به یک مسیر واحد بطور مثال مقصد داریم آدرس Shop.aftab.cc و www.Shop.aftab.cc و دیگر مسیرهایی که به پوشه Shop هدایت می‌شود را به http://aftab.cc/shop هدایت کنیم. و یا انتقال کاربران از دامنه‌های مختلف به یک دامنه . و یا هدایت کاربران از حدسیات مختلف آن‌ها به مسیری که انتظار دارند . مثلا می‌خواهیم کاربران از مسیرهای Mobile.aftab.cc و یا M.aftab.cc را به نسخه‌ی موبایلی سایت هدایت کنیم. برای انجام این کار در فایل htaccess بنویسید:

```
Redirect/مسیر قدیمی مسیر جدید http://
```

ابزار Redirect در کنترل پنل Cpanel این کار را انجام می‌دهد.

شکل ۳-۱۰ ابزار Redirect در کنترل پنل Cpanel

Wildcard در Redirect به چه معناست؟ در صورتی که در Redirect امکان Wildcard فعال باشد بخشی از انتهای آدرس قدیمی که در Redirect بیان شده، جدا شده و به انتهای آدرس قبلی می‌چسبد. اگر کاربر وارد کند aftab.cc/v.php و تنظیم شده باشد که aftab.cc به aftab.cc هدایت می‌شود

شکل ۳-۱۱ Redirect کردن در کنترل پنل Cpanel

- **Password protection[1]:** می‌توان بروی پوشه‌ها و فایل‌های مختلف سایت. با استفاده از قطعه کد زیر که در htaccess تایپ می‌شود پسورد گذاشت.

عنوان پنجره:

```
AuthName"Member's AreaName"
```

فایل‌های حاوی یوزرنیم و پسوردهای مجاز:

```
AuthuserFile/path/to/password/file/.htaccesswd
AuthType Basic
require valid-user
```

نکته۱: ترجیا فایل حاوی پسوردها را در سطحی قبل از سطح www قرار دهید.

[1] حفاظت با پسورد

نکته۲: بطور پیش‌فرض فایل حاوی یوزرنیم و پسوردها با نام Passwd و در پوشه‌ای بنام htpasswds. در یک سطح بالاتر از www ایجاد می‌شود.

نکته ۳: پسوردها باید به صورت MD5 شده در فایل حاوی یوزرنیم و پسوردها به صورت زیر قرار دهید.

نکته۴: گزینه‌ی password protect directory برای این‌کار تعبیه شده است.

تمرین: در هاست سایت خود یک فایل بنام admin.php ایجاد کنید و با htaccess روی آن پسورد بگذارید.

پسورد گذاشتن روی یک فایل با htaccess:

شکل ۱۲-۳ مسیر پسورد گذاشتن روی فایل‌ها و پوشه‌ها در htaccess

برای این‌کار قطعه کد به صورت زیر تغییر می‌کند:

```
AuthUserFile /.htpasswd
AuthName "Name of Page"
AuthType Basic
<Files "mypage.html">     نام و آدرس فایل موردنظر
  require valid-user
</Files>
```

نکته۵: فایل htaccess به حروف بزرگ و کوچک حساس است.

نکته۶: فایل htaccess بسیار فایل حساسی است و کوچکترین خطا در نگارش قوانین باعث خطای ۵۰۰ می‌شود.

- **Deny visitors by IP address[1]:** با استفاده از ساختار زیر می‌توان IP یا IPهای خاصی را به روی پوشه یا سایت ممنوع یا مجاز کرد.

```
Order allow,deny
Deny from 255.0.0.0
Deny from 123.456.
Allow from all
```

کلمه‌ی کلیدی order ترتیب اجرای قوانین allow و deny را مشخص می‌کند ممکن است بخواهیم سایت برای همه باز باشد و فقط برای آی پی های خاصی بسته باشد در این صورت order به صورت allow,deny خواهد بود. ولی ممکن است بخواهیم سایت بروی همه بسته باشد و فقط افراد خاصی مجاز باشند در این صورت order به صورتdeny,allow خواهد بود.

مسئله1: فایل htaccess را طوری تنظیم کنید که فقط خودتان که IPتان ۲.۳.۴.۵ است، دسترسی داشته باشید.

```
Order deny,allow
Allow from 2.3.4.5
Deny from all
```

مسئله2: طوری تنظیم کنید که رنج IPهای ۲.۳.۴.۱ تا ۲.۳.۴.۱۰۰ که از یک Isp خاص است، به سایت دسترسی نداشته باشد.

```
Order deny,allow
Allow from all
Deny from ۲,۳,۴,۱-2.3.4.100
```

مسئله۳: تنظیم کنید IPهای اینترنتی ایران(رنج ۱۰) بتوانند وارد سایت شوند و بقیه نتوانند.

```
Order deny,allow
Allow from 100.0.0.0/8
Deny from all
```

[1] ممنوع کردن دسترسی بازدید کننده‌ها از طریق آی پی آن‌ها

گزینه Ip address member در کنترل پنل Cpanel برای اینکار تعبیه شده است.

- **Deny visitors by referrer:** ممنوع کردن دسترسی کاربرانی که با کلیک روی سایت ما در سایت‌های دیگر وارد سایت می‌شود. ممکن است بخواهیم ورودی‌های سایت که از سایت خاصی وارد شده‌اند، که در اصطلاح به آن سایت Referrer[1] گفته می‌شود را، بلوکه کنیم. با استفاده از قطعه کد زیر می‌توان این کار را انجام داد.

```
RewriteEngine on
#options +FllowSymlinks
RewriteCond%{HTTP_REFERRER}otherdomain\com[NC,OR]
RewriteCond%{HTTP_REFERRER}anotherdomain\com
RewriteRule.*-[F]
```

نکته: NC در قطعه کد بالا به معنی not case-sensitive بوده یعنی آدرس Referrer به حروف بزرگ و کوچک حساس نیست.

چرا ممکن است بخواهیم Referrerها را بلوکه کنیم؟
احتمال اول: ممکن است یک سایت غیر اخلاقی برای افزایش بازدید سایت خودش به سایت شما لینک بدهد تا در لیست Referrerهای سایت شما قرار گیرد.

احتمال دوم: سایت‌های Download برای جلوگیری از کپی غیرمجاز لینک فایل‌های خود را در سایت‌های دیگران، Referrerها به پوشه حاوی فایل‌ها را می‌بندند تا کاربران مجبور شوند فقط بروی سایت خودشان بروی لینک‌ها کلیک کنند.

احتمال سوم: ممکن است یک لینک به فایلی در سایت شما در سایت‌های پر بازدید قرار گیرد و به نوعی حمله‌ی DOS بروی سرور شما شکل بگیرد. برای جلوگیری از ورود کاربران از آن Referrer می‌توانید آن سایت را بلوکه کنید.

[1] اشاره کننده

- **Hotlink prevention[1]:** اگر یک وب‌سایت به منبعی در سایت شما(بطور مثال به فایل‌های gif,jpg,css یا js) بطور مستقیم دسترسی داشته باشد، می‌تواند در سایت خود از آن منبع استفاده(سوءاستفاده) کند و باعث می‌شود ترافیک شما بدون هیچ سودی برای شما استفاده شود. به این نوع استفاده از منابع سایت‌های دیگر در یک سایت در اصطلاح Hotlink گفته می‌شود. برای جلوگیری از این سوءاستفاده می‌توانید در فایل htaccess قطعه کد زیر را درج کنید.

```
RewriteEngine on
RewriteCond%{HTTP_REFERRER}'^$
RewriteCond%{HTTP_REFERRER}'^http://yourdomain.com/.*$[NC]
RewriteRule\(gif\jpg\css)$-[F]
```

این قطعه کد به وب‌سرور آپاچی می‌گوید همه‌ی لینک‌ها به فایل‌های gif,jpg,css که از دامنه‌ی yourdomain.com نیست را بلوکه کن.(**توجه**: بجای yourdomain.com باید دامنه و پسوند سایت خود را وارد کنید.). با استفاده از قطعه کد زیر می‌توان بجای عکس‌های gif,jpg که بصورت hotlink استفاده شده‌اند یک عکس دلخواه، یک گلایه نامه یا یک تبلیغات قرار داد.

```
RewriteEngine on
RewriteCond%{HTTP_REFERRER}'^$
RewriteCond%{HTTP_REFERRER}'^http://(www\)?yourdomain.com/.*$[NC]
RewriteRule\(gif\jpg)$ http://www. yourdomain.com/hotlink.jpg[R,L]
```

کاربرد برخی نمادها در فایل htaccess:

!	به معنی not
^	به معنی شروع شد با
()	اختیاری بودن
؟	به معنی صفر یا یک کاراکتر
*	به معنی صفر تا بینهایت کاراکتر
$	پایان پذیر با
|	<< یا >>
\	برای اسکیپ کردن کاراکترهای خاص

[1] جلوگیری از لینک مستقیم به منابع

[]	بازه‌ای از اعداد یا بازه‌ای از کاراکترها [A-Z]

گزینه hotlink protection در کنترل پنل Cpanel برای اینکار تعبیه شده است.

شکل ۳-۱۳ گزینه hotlink protection در کنترل پنل Cpanel

- **کاربردهای مربوط به Directory index[1]**: دستور directory index اجازه می‌دهد تا صفحه پیش فرض مشخص کنید تا اگر کاربر یک پوشه را فراخوانی کرد به جای لیست محتوای آن پوشه یا فایل پیش فرضی که در وب‌سرور فایل‌های موردنظر شما فراخوانی شود. برای این‌کار از قطعه کد زیر در فایل htaccess استفاده کنید.

```
DirectorIndex  index.html
```

با استفاده از قطعه کد زیر می‌توان اولویت فایل‌های index را تعیین کرد.

```
DirectorIndex  index.html index.cgi index.php
```

- **htaccess directory listing[2]:** یکی از راه‌های امنیت سایت جلوگیری از نمایش فایل‌های داخل یک پوشه است. برای این کار می‌توان در فایل htaccess نوشت:

```
IndexIgnore*
```

این قطعه کد یعنی لیست کردن همه‌ی محتوای پوشه ممنوع. در صورتی که نیاز دارید فقط فایل‌های خاص لیست شوند از مثال زیر استفاده کنید.

```
IndexIgnore*.zip*.jpg*.gif
```

[1] تعیین صفحه پیش فرض لیست محتوای پوشه

[2] غیر فعال لیست محتوای پوشه

برای فعال کردن امکان ایست محتوای پوشه‌ها (در صورتیکه بطور پیش‌فرض توسط وب‌سرور غیرفعال شده باشد) از قطعه کدزیر استفاده کنید.

```
Options+Indexes
```

از قطعه کد زیر می‌توان برای غیر فعال کردن لیست شدن محتوا استفاده کرد.

```
Options-Indexes
```

- **Settings Server Time zone[1] :** با استفاده از قطعه کد زیر می‌توان تایمزون سایت یا وب‌سرور زیرا به منطق‌ی موردنظر خود تنظیم کرد.

```
SetEnv Tz Asia/Tehran
```

- **Prevent access to php.ini** :جلوگیری از دسترسی فایل‌های خاص سایت فایل مانند فایل php.ini،یا فایل htaccess. با استفاده از قطعه کدزیر می‌توان از دسترسی کاربران با فایلی به نام php.iniجلوگیری کرد.

```
<files "php.ini" >
Order allow,deny
Deny from all
</files>
```

برای بستن فایل htaccess بجای php.ini بایدhtaccess. بنویسیم.

- **Ensuring media files are downloaded instead of played :** جلوگیری از پخش فایل در مرورگر و اجبار به دانلود آن بجای پخش. قطعه کد زیر از بازشدن مستقیم فایل‌های zip,Mp3,Mp4 در مرورگر کاربر جلوگیری می‌کند.

```
AddType application/octet-stream.zip.mp3.mp4
```

- **Preventing requests with invalid characters:** جلوگیری از تقاضای حاوی کارکترهای غیر مجاز. قطعه کد زیر url هایی که حاوی کارکترهای غیر مجاز (مانند ٪ یا " " یا ‘ ‘ یا $) باشد را غیرقابل فراخوانی می‌کند.

```
RewriteEngine on
RewriteBase/
RewriteCond %(THE_REQUEST)!^[A-Z]{3,9}\[a-zA-Z0-9\.\+_-\?\=\&]+\HTTP/[NC]
RewriteRule.*-[F,NS,L]
```

[1] تنظیم کردن time zone سرور

Session hijacking[1] 7-2-3: یکی از حملات نوع man in the middle "حمله‌ی سرقت جلسه" است. در این نوع حمله شناسه‌ی جلسه‌ی کاربر یا دستگاهی که به سرور متصل شده توسط نفوذگر دزدیده شده و به‌عنوان شناسه‌ی خود نفوذگر استفاده می‌شود.

تمرین: عملا حمله‌ی Session hijacking را پیاده‌سازی کنید.

روش انجام حمله سرقت جلسه : هر کاربری که به وب‌سروری متصل می‌شود وب‌سرور برای شناسایی در مراجعات بعدی به‌طورخودکار یک کوکی بنام Session ID (یا نام‌های مشابه) روی سیستم کاربر ایجاد می‌کند، که محتوای آن یک عدد Random بسیار بزرگ است. حمله‌ی سرقت جلسه فقط در مورد سایت‌هایی اتفاق می‌افتد که برای شناسایی کاربر از session استفاده می‌کنند و نه کوکی (مانند سایت‌های بانک‌ها، سرویس‌های ایمیل و ...) اگر بتوان محتوای کوکی Session ID را از سیستم قربانی به طریقی (مانند packet sniffing و یا سرقت بصورت حضوری و یا حمله به سرور سایت) از سیستم او بدست آوریم .و جایگزین محتوای Session ID در سیستم خود بعنوان نفوذگر کنید.آن سایت از این پس مارا به‌عنوان صاحب آن Session ID می‌شناسد. بنابراین اگر کاربران در آن سایت به‌عنوان یک کاربر مجاز لاگین کرده باشد، مانیز پس از این سرقت، یک کاربرِ مجازِ لاگین شده به حساب می‌آییم.

نکته: این نوع حمله باید در بازه زمانی‌ای که کاربر در سایت موردنظر حاضر است، اتفاق بیافتد و به محض بستن مرورگر امکان حمله از بین خواهد رفت.

۳-۲-۸ اقدامات متقابل در برابر Session hijacking :

- با استفاده از پروتکل‌هایی مانند SLL ارتباطات جلسه را رمزنگاری کنید.
- از کانال‌های ارتباطی رمزنگاری شده مانند VPN استفاده کنید. (VPN رمزنگاری شده منظور است.)
- Patchهای امنیتی مربوط بهتهدیدات امنیتی مربوط به TCP/IP بر روی سیستم کلاینت و سرور نصب کنید.

تحقیق: تحقیق کنید که از خطر امنیتیِ Predictable packet sequences چگونه می‌توان استفاده کرد.

تحقیق: چگونه HTTPS از Session hijacking جلوگیری می‌کند؟

[1] دزدی جلسه

Denial of service[1] **9-2-3**: ممکن است نفوذگر با استفاده از حملاتی مانند SYN Flood[2] که یک حمله‌ی DOS در شبکه به حساب می‌آید[3]. دسترسی کاربران قانونی[4] به سرویس‌ها را انکار کند. در این نوع حملات بیش از ظرفیت سرور، تقاضا به آن ارسال می‌شود. نفوذ معمولاً از طریق تهدیدات بالقوه در ارتباط TCP/IP اتفاق می‌افتد و سرور را با صفی از ارتباطات در حال انتظار مواجه می‌کند.

۱۰-۲-۳ اقدامات متقابل در برابر حمله Denial of service :

- آخرین سرویس‌پک‌ها را برروی پلت‌فرم‌ها نصب کنید.
- تنظیمات رجیستریِ بسترها طوری تنظیم کنید که صفِ ارتباطات TCP افزایش پیدا کند و یا مدت زمان اتصال یک کلاینت[5] به سرور کاهش پیدا می‌کند.
- از مکانیزم‌های ثبت ردپا ی پویا برای اطمینان از اینکه در صف ارتباطات هرگز سریز رخ نمی‌دهد استفاده کنید.
- از سیستم‌های تشخیص ورود سرزده و غیرمجاز به شبکه که به صورت خودکار حملات SYN را تشخیص می‌دهد در شبکه استفاده نمایید.

تحقیق: در مورد Intrusion Detection System(IDS) تحقیق کنید.

قطعه کد زیر می‌تواند در PHP تاحد زیادی از Session Hijacking بکاهد.

```
۱   <?php
۲   Ini_set('SESSION.cookie_httppnly',true);
۳   Session_start();
۴   If(isset($_SESSION['user_last_ip'])===false)
۵   {
۶     $_SESSION['user_last_ip']= $_SERVER['REMOTE_ADDR'];
۷   }
۸   If($_SESSION['user_last_ip']!== $_SERVER['REMOTE_ADDR'])
۹   {
۱۰      Session_unset();
```

[1] خود داری از خدمات
[2] طوفان
[3] Network level denial of service attack
[4] Legitimate users
[5] Connection establishment period

```
۱۱      Session_destroy();
۱۲  }
۱۳  ?>
```

تحلیل کد بالا:

- در خط ۴ بررسی می‌شود که آیا سشنی بنام User_ last_ip برای این کاربر set شده یا خیر، اگر set شده باشد یعنی اولین حضور کاربر در سایت است .
  - پس در خط ۶، IP فعلی کاربر(REMOTE-ADDR)درسشن User_Last_ip ذخیره می‌شود.
  - در خط ۸ که در مراجعات بعدی کاربر فعال خواهد شد، بررسی می‌شود که اگر IP فعلی کاربر (REMOTE-ADDR) با IP که در سشن User_Last_ip ذخیره شده (یعنی در مراجعه قبلی او) برابرنبود یعنی سشن توسط یک کاربر دیگر سرقت شده است .
  - بنابراین در خط ۱۰ و ۱۱ کلیه‌ی سشن‌های کاربرپاک شده و در حقیقت هر دو کاربر)قربانی و هکر(logoff خواهد شد.

نکته: این قطعه کد در برابر حملاتی که هکر و قربانی در یک شبکه‌ی واحد)مانند کافی‌نت یا سایت‌های عمومی(قرار داشته باشند موثر نخواهد بود چرا که IP هردو از نگاه سرور سایت یکی است.

تحقیق: چگونه می‌توان از حملات Session hijacking در شبکه‌های عمومی نیز جلوگیری کرد؟

فصل چهارم

۴-۱ تهدیدات مربوط به هاست و اقدامات متقابل آن: تهدیدات Host بر روی "نرم‌افزارسیستمی" که برنامه شما برروی آن ساخته و اجرا شده تأثیر می‌گذارد، نرم‌افزارهای سیستمی مانند: ویندوز سرور، IIS[1]، .NET Fram work ، SQL Server ویا یک Role خاص بر روی یک سرور.

۴-۱-۱ مهمترین تهدیدات این سطح عبارتتند از:

- **Viruses**[2]: ویروس برنامه‌ای است که برای انجام عملیات مخرب و ایجاد اختلال در سیستم‌عامل یا برنامه‌ی شما طراحی شده است.
- **Trojan horses**[3]: اسب تروجان مانند یک ویروس عمل می‌کند با این تفاوت که کد مخرب، داخل فایل داده‌ای یا برنامه‌ی اجرایی که بنظر می‌رسد، بدون ضرر باشد، مخفی می‌شود.
- **worms**[4]: کرم مانند اسب تروجان عمل می‌کند با این تفاوت که خودش را از یک سرور به سرور دیگر تکثیر می‌کند[5]. کرم‌ها معمولاً به سختی تشخیص داده می‌شوند چرا که بطور منظم فایل‌هایی که بتوان دید، ایجاد نمی‌کنند. آن‌ها معمولاً زمانی شناسایی می‌شوند که متوجه شویم منابع سیستم بیش از حد مصرف می‌شود و یا اجرای برخی برنامه‌های همیشگی متوقف شده است.

بطور مثال code Red worm مطرح‌ترین کرمی است که IIS را تحت تأثیر قرارمی‌دهد و موجب یک سریز در سرور می‌شود.

موفقیت این نوع حملات وابسته به نقاط آسیب‌پذیری مختلفی مانند: پیش‌فرض‌های ضعیف ، باگ‌های نرم‌افزاری ،خطاهای سمت کاربر و آسیبب‌پذیری‌های به ارث رسیده از پروتکل‌های اینترنتی است.

۴-۱-۲ اقدامات متقابل علیه این نوع حملات:

- پورت‌های غیرلازم را بر روی فایروال و هاست بلاک کنید.
- پروتکل‌ها سرویس‌هایی که از آن استفاده می‌شود غیرفعال کنید.
- تنظیمات پیش‌فرض و ضعیف سرور را محکم کاری کنید.

---

[1] Internet Information services
[2] ویروس
[3] اسب تروجان
[4] کرم‌ها
[5] self-replicates

تمرین: یک کد مخرب دریک صفحه وب قرار دهید که با فراخوانی آن در تمام پوشه های سایت فایلی با نام hack.php ایجاد شود، که درآن فایل‌ها یک حلقه‌ی بی‌نهایت در حال شمارش باشد و به طریقی آن فایل‌ها را اجرا کنید تا منابع وب‌سرور اشغال شود.

3-1-4 تهدید Footprinting در سطح هاست: تکنیکی برای جمع‌آوری اطلاعات در مورد سیستم‌های کامپیوتری و دیگر موجودیت‌ها، که با آن سیستم‌ها مرتبط هستند، است. این‌کارازطریق به کار گرفتن تکنیک‌های مختلف مانند ترفندهای زیر انجام می‌شود:

- DNS queries
- Network enumeration
- Network queries
- Operating system identification
- Organizational queries
- Ping sweeps
- Point of contact queries
- Port Scanning
- Register queries(WHOIS queries)
- SNMP queries
- World Wide Web spidering

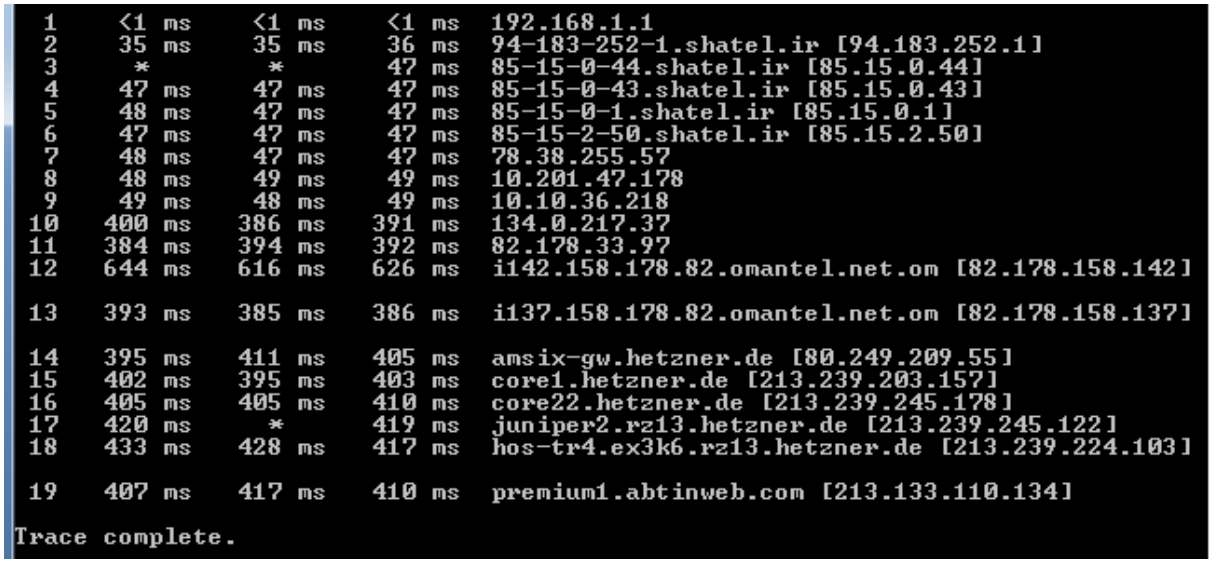
۱- بطورمثال با استفاده از دامنه‌ی یک سایت می‌توان به IP سروری که آن سایت برروی آن قرار دارد پی برد.

```
Pinging aftab.cc [213.133.110.134] with 32 bytes of data:
Reply from 213.133.110.134: bytes=32 time=429ms TTL=47
Reply from 213.133.110.134: bytes=32 time=439ms TTL=47
Reply from 213.133.110.134: bytes=32 time=442ms TTL=47
Reply from 213.133.110.134: bytes=32 time=413ms TTL=47
```

شکل ۴-۱ Ip سرور سایت مورد نظر

۲- یک فعالیت محاسباتی است که usernameها و اطلاعات گروه‌های منابعِ به اشتراک گذاشته شده و سرویس‌های کامپیوترهای درون شبکه، جمع‌آوری می‌شود. نرم‌افزارهایی مانند Nmap و Nessus در این زمینه کاربرد دارد.

۳- بطورمثال با استفاده از از دستورTracert aftab.cc لیست روترها و مقاصدی که بین کامپیوترِ کاربر و سرور سایت aftab.cc است را نمایش می‌دهد.



شکل ۴-۲ لیست روترها و مقاصدی که بین کامپیوترِ کاربر و سرور سایت aftab.cc است

۴- بطورمثال با استفاده از . نرم‌افزار Nmap و یا با استفاده از افزونه‌هایی مانند Wappalyzer در فایرفاکس می‌توان به نام سیستم‌عامل سرور و کتابخانه‌های بکاررفته در سایت پی برد.

تمرین: این افزونه را روی فایرفاکس نصب کنید و مشخص کنید سایت Irna.ir از چه سیستم‌عامل و چه کتابخانه‌هایی استفاده می‌کند.

۵- بطورمثال با استفاده از دستور Ping aftab.cc می‌توان به سرعت و مسافت بین کاربر و سرور سایت aftab.cc پی برد.

۶- بطورمثال با استفاده از اطلاعات موجود در WHOIS دامنه می‌توان شرکت پشتیبان سایت و یا هاستینگ سایت را شناسایی کرد و یا در صورت لزوم با آن‌ها تماس گرفت.

۷- همان‌طور که توضیح داده شد با استفاده از نرم‌افزارهایی مانندAdvanced port scanner، می‌توان به پورت‌های آسیب‌پذیر یک سرور پی برد.

۸- با استفاده از سایت‌های WHOIS می‌توان اطلاعاتی در مورد صاحب دامنه یا IP به دست‌آورد.
نمونه‌ای از سایت‌های WHOIS:

- WHOIS. Sc/
- DNSstuff.com

۹- با استفاده از لینک‌های داخل یک سایت و یا سایت‌هایی که به سایت لینک داده‌اند می‌توان اطلاعاتی در مورد سایت جمع‌آوری کرد.

SNMP چیست؟ مخفف Simple Network Management Portocol و به معنیپروتکل مدیریت آسان شبکه یک پروتکل استاندارد اینترنتی برای انتساب دستگاه‌ها به IP ها در یک شبکه‌ی مبتنی برIP است. دستگاه‌هایی مانند: روترها، سوئیچ‌ها، سرورها، کلاینت‌ها و ...

4-1-4 اقدامات متقابل در برابر Footprinting:

- پروتکل‌های غیرضروری را غیر فعال کنید.
- پروت‌های خطرناک بلاک کنید.
- از پروتکل‌های امنیتی مانند IPsec[vii] و فیلترهای.TCP/IP استفاده کنید
- IIS را طوری تنظیم کنید که راه‌های جمع‌آوری اطلاعات از آن بسته باشد.
- از سیستم‌های ضد نفوذ[1] برای ریجکت کردن ترافیک‌های خطرناک که با الگوهای Footprinting می‌خوانند، استفاده کنید.

نمونه سوال: Footprinting چیست، ۴ مورد از تکنیک‌های آن را با ذکر مثال توضیح دهید و اقدامات متقابل در برابر آن را نام ببرید.

5-1-4 Password Cracking[2]: اگرنفوذگر نتوان یک اتصال ناشناس[3] با سرور برقرار کند تلاش می‌کند که یک اتصال شناخته شده[4] برقرارکند. برای این‌کار نفوذگرباید یک یوزرنیم و پسورد معتبر را بداند. اگر شما از یوزرنیم‌های پیش‌فرض در یک سیستم (مانند admin,root و...) استفاده کرده باشید، نقطه‌ی شروع را برای نفوذگر مشخص کرده‌اید. بنابراین او فقط کافی است به سراغ پسورد برود، استفاده از پسوردهای ضعیف و یا حتی گاهی خالی (Blank)کار را برای نفوذگر از این‌هم آسان‌تر می‌کند.

6-1-4 اقدامات متقابل در برابر Password Cracking[5]:

- از پسوردهای قوی[6] برای همه نوع اکانت استفاده کنید و کاربران را نیز مجبور کنید پسوردهای قوی انتخاب کنند (پسورد قوی و امن پسوردی است که اولاًحداقل ۸ کارکتر باشد و ثانیاً سه مورد از موارد زیر در آن بکاررفته باشد:
- حروف بزرگ انگلیسی
- حروف کوچک انگلیسی

---

[1] IDS
[2] شکستن پسورد
[3] A nonymous connection
[4] Outhentication connection
[5] شکستن پسورد
[6] strong

- اعداد
- حروف غیرالفبایی (&-$_!))
- سیاست‌های Lockout برای اکانت‌ها در نظربگیرید تا تعداد تلاش‌های ناموفق برای حدس پسورد را محدود کنید.(بطورمثال اکثرکنترل پنل‌های سایت‌ها با ۵ بار تلاش ناموفق برای شکستن پسوردIP نفوذگر را تا چند ساعت در فایروال سرور بلاک می‌کنند و یا در ویندوز سرور یک Policy یا سایت Group policy وجود دارد که می‌توان تعداد تلاش‌های ناموفق و مدت زمان قفل شدن اکانت را تنظیم کرد.
  - از نام‌های کاربری پیش‌فرض و یا قابل حدس مانند administrator,admin و غیره استفاده نکنید.
  - لاگین‌های ناموفق را Audit[1] کنید تا در صورت لزوم اطلاعاتی در مورد نفوذگر داشته باشید.

7-1-4 حملات Denial of service : حملات DOS می‌تواند با روش‌های مختلفی در زیر ساخت شما صورت بگیرد . مثلاً در هاست نفوذگر می‌تواند با استفاده از حمله‌ی bruteforce که روی برنامه‌ی شما انجام می‌دهد سرویس‌دهی را مختل کند ویا با استفاده از آن به برنامه یا سیستم‌عامل سرور، نفوذ کند.

8-1-4 اقدامات متقابل علیه این نوع حمله:

- برنامه نوشته شده توسط خودش و سیستم‌عامل و سرویس‌ها در برابر پیکربندی سند و در ذهن داشته باشید این حمله در سه سطر رخ می‌دهد. ایمن‌سازی در یک سطر کافی نیست.
- آخرین آپدیت‌ها و patcheهای امنیتی در هر سه سطر نصب کنید.
- با پروتکل‌های TCP/IP در برابر IIS محکم کاری کنید.
- مطمئن شوید که از سیاست Lockout نمی‌توان در برابر نام کاربری‌های مهم‌تر(مانند: نام کاربری مدیر سایت) سوءاستفاده کرد. بطور مثال: اگر تنظیم کرده باشید که نام کاربری admin با ۵ بار پسورد اشتباه قفل شود، کاربران می‌توانند براحتی برای admin مشکل ایجاد کنند.
- مطمئن شوید که برنامه‌ی شما برای برخورد با لودهای زیاد و غیرطبیعی بهینه سازی شده است.
- عملکرد برنامه‌تان را در صورت مواجه شدن با شکست[2] بررسی کنید.
- از یک سیستم IDS برای تشخیص حملات DOS بالقوه استفاده کنید.

---

[1] ثبت ردپا
[2] شکست

9-1-4 حمله Arbitrary Code Execution[1]: ممکن است نفوذگر یک کد مخرب را برروی هاست یعنی سرور شما، اجرا کند و از این طریق منابع سرور را مشغولِ آن کد کند. پیکربندی ضعیف IID و نصب نکردن بسته‌های امنیتی می‌تواند موجب این حملات وسریز بافر و دسترسی به مسیرهای غیرمجاز شود. (path traversal: اگر نفوذگر بتواند به مسیرهایی که به او اجازه داده شده دسترسی پیدا کند.)

10-1-4 اقدامات متقابل در برابر این نوع حمله:

- پیکربندی وب‌سرور برای ممنوع کردن URL های حاوی "/.." (یعنی یه سطح بیاد عقب)
- جلوگیری از اجرای برنامه‌ها و اجرای دستورات سیستمی از طریق ACL[2] ها .
- جلوگیری از آپلود و اجرای فایل‌های exe و bat(batch)[viii] از طریق فیلترکردن آن‌ها در IIS به‌ویژه در سیستم‌عامل ویندوز.
- آخرین وصله‌های امنیتی و آپدیت‌ها را روی سرور نصب کنید.

11-1-4 خطر امنیتی Aunauthorized Access[3]: تنظیمات نادرست مربوط به دسترسی ممکن است منجر به دسترسی نفوذگر به اطلاعات حساس و امنیتی و یا اجرای کارهای خطرناک شود. ضعف‌های تنظیمات IIS و دسترسی‌های NTFS از جمله دلایل ایجاد این نوع تهدید هستند.

12-1-4 اقدامات متقابل در برابر این نوع تهدید:

- دسترسی‌های مختلف در فایل‌ها و پوشه‌های سایت را درست تنظیم کنید .
- فایل‌ها و پوشه‌هایی که نباید کاربران خاص به آن‌ها دسترسی داشته باشند، از طریق دسترسی‌های NTFS محدود کنیم.
- با استفاده از فریم‌ورکِ NET. برنامه‌های Asp.Net خود را از لحاظ دسترسی به محل‌های غیرمجاز امن کنید.

تحقیق: امکانات موجود در Net framework. برای ایمن‌سازی برنامه‌های Asp.Net چیست؟

---

[1] اجرای کد دلخواه
[2] Access Control List
[3] دسترسی غیر مجاز

فصل پنجم

۵-۱ تهدیدات در سطح Application و اقدامات متقابل آن: برای آنالیز بهتر تهدیدات سطح برنامه، بهتر است آن‌ها را از نظر نوع تهدید دسته‌بندی کنید .جدول زیر خلاصه‌ای از تهدیدات سطح برنامه در دسته‌بندی های مختلف را نشان می‌دهد:

• تهدیدات مربوط به اعتبارسنجی[۵] ورودی به برنامه	- Buffer overflow[1] - Cross sit Scripting[2] - SQL injection[3] - Canonicalization[4]
• تهدیدات مربوط به اهراز هویت[۸]	- Network eavesdroping[6] o Brute force attacks o Dictionarry attaks o Cookie replay o Credential theft[7]
• تهدیدات مربوط به سطح دسترسی‌ها[۱۳]	o Elevation of privilege[9] o Disclosure of confidential data[10] o Data tampering[11] o Luring attaks[12]

[1] سریز بافر
[2] تزریق اسکریپت از سایت دیگر
[3] تزریق کد اس کیو ال
[4] متعارف سازی ، قانونی سازی
[5] Input Validation
[6] استراق سمع شبکه
[7] سرقت عامل اهراز هویت
[8] Authentication
[9] ارزیابی سطح دسترسی
[10] افشای اطلاعات محرمانه
[11] دستکاری داده
[12] حملات برگشتی
[13] Authorization

• تهدیدات مربوط به مدیریت پیکربندی‌ها و تنظیمات[۷]	○ unauthori zed access to administration interface[1] ○ Unauthorized access to configuration stores[2] ○ Retrieval of clear text configuration data[3] ○ Lack Of individual accountability[4] accountability[5] ○ Over privileged process and service accounts[6]
• خطرات مربوط به داده‌های حساس[۱۱]	○ Access sensitive data in storage[8] ○ Network eavesdropping[9] - Data tampering[10]

[1] دسترسی غیر مجاز به واسطه‌های مدیریتی
[2] دسترسی غیرمجاز به مخازن تنظیمات
[3] بازیابی متن واضح(رمزنگاری نشده)مربوط به داده‌های پیکربندی
[4] عدم وجود اکانت برای هر شخص با مسئولیت مختص به خود
[5] نبود اکانت‌های شخصی برای افراد
[6] ایجاد اکانت‌ها و فرآیندهایی که بیش از حد لازم دسترسی دارند
[7] Configuration Management

[8] دسترسی به داده‌های حساس حافظه
[9] استراق سمع از طریق شبکه
[10] دستکاری داده
[11] Sensitive data

• تهدیدات مربوط به مدیریت سشن[۳]	○ Sessin hijacking[1] ○ Session replay[2] - Man in the middel
• تهدیدات مربوط به رمزنگاری[۶]	○ Poor key generation or key management[4] ○ Weak or custom encryption[5]
• تهدیدات مربوط به دستکاری پارامترهای ردو بدل شده با سرور[۱۱]	○ Query string maipulation[7] ○ Form filed manipulation[8] ○ Cookie manipulation[9] ○ HTTP header manipulation[10]

[1] سرقت جلسه
[2] استفاده مجدد از سشن
[3] Session management
[4] تولید و مدیریت ضعیف کلید رمزنگاری
[5] رمزنگاری ضعیف
[6] Cryptography
[7] دستکاری کوئری ها
[8] دستکاری فیلدهای فرم
[9] دستکاری کوکی
[10] دستکاری هدر Http
[11] Parameter manipulation

• تهدیدات مربوط به مدیریت استثنا[3]	○ Information disclosure[1] - denial of service[2]
• زیرنظرگرفتن وثبت ردپا[7]	○ User denies performing an operation[4] ○ Attacker exploits an appli9cation without trace[5] ○ Attacker covers his or her tracks[6]

1-1-5 اعتبارسنجی ورودی‌ها به برنامه: وقتی نقاط آسیب‌پذیر سطح هاست و شبکه بطور کامل امن باشد (که اکثر اوقات به همین صورت است) برنامه شما، به تنها راه نفوذ برای یک هکر تبدیل خواهد شد. ورودی‌های برنامه شما یک واسط برای تست سیستم شما و همین‌طور راهی برای اجرای کد از طرف هکر است. اگر برنامه‌ی شما به صورت کورکورانه[8] به هرنوع ورودی اعتماد کند این می‌تواند از اولین خطرات ایجادشده در برنامه باشد.

حملات زیر از طریق ورودی به برنامه‌ی شما انجام می‌شود:

- **Buffer overflows:** سریز بافر می‌تواند منجر به حملات DOS یا تزریق کد شود. همان‌طور که گفته شد حمله‌ی DOS باعث CRASH شدن فرآیند می‌شود و حمله‌ی Code injectionروال اجرای برنامه‌ی شما طوری تغییر می‌دهد، که کدِ مخربِ هکر اجرا شود.

تمرین: مثال زیر نمونه‌ای از حمله‌ی سریز بافر در php است:

[1] افشای اطلاعات
[2] امتناع از سرویس
[3] Exception management
[4] کاربر انجام یک عملیات را انکار می‌کند
[5] نفوذگر بدون اینکه ردپایی از خود به جا بگذارد به برنامه نفوذ می‌کند
[6] نفوذگر ردپای خود را می‌پوشاند
[7] Auditing and logging

[8] Blindly

```
Void SomeFunction( char *pszInput )

{

Char szBuffer[10];

// Input is copied straight into the buffer when no type checking is
performed strcpy(szbuffer, pszInput);

. . .

}
```

قطعه کد بالا شمای کلی سریز بافر را نشان می‌دهد. در این کد یک ورودی به یک تابع داده شده است و آن ورودی بدون بررسی نوع و اندازه‌ی آن به بخشی از بافِر کپی شده است. هکِر می‌تواند به‌عنوان ورودی یک داده‌ی حجیم‌تر از ظرفیت بافر(که در اینجا ۱۰ خانه است) تزریق کند و موجب سریز بافر شود.

- **اقدامات متقابل در برابر این حمله:**
- کل ورودی‌های برنامه را اعتبارسنجی کنید این اولین راه حفاظت در برابر این نوع حمله است. به‌ویژه فیلدهای آپلودِ فایل باید به‌طور دقیق مورد بررسی قرار گیرد.
- ورودی‌ها باید از لحاظ نوع داده، طول، فرمت(ساختار ورودی)، رِنج(محدوده‌ی معتبر ورودی) بررسی شود.
- تنها به اعتبارسنجی سمت کلاینت یا سرور اعتماد نکنید، بلکه در هردو طرف کار اعتبارسنجی را با قدرت انجام دهید.
- تا حد ممکن از اسکریپت‌ها و API های مدیریت نشده در برنامه‌ی خود استفاده نکنید.
- مثالی از تزریق کد از طریق سریزِ بافر: ممکن است نفوذگر در بخشی از برنامه که مسئولیت اعتبارسنجی را بر عهده دارد از طریق حملات DOS یا ورودی نامعتبر کاری کند در آن بخش از برنامه سریز رخ دهد و از کار بیافتد. بنابراین نفوذگر می‌تواند اعتبارسنجی را دور زده و کد دلخواه خود را تزریق کند.

تمرین: یک صفحه‌ی HTML که با جاوا اسکریپت اعتبار کد ملی را تأیید کند، طراحی کنید و در سمت سرور نیز با PHP کار اعتبارسنجی را انجام دهید. نشان دهید. که نمی‌توان باغیرفعال کردن JS سیستم را دورزد.

- **حمله Cross site scripting**: یک حمله‌ی XSS می‌تواند موجب شود که یک قطعه کد دلخواه در مرورگر کاربر اجرا شود در حالی که مرورگر به یک سایت قابل اعتماد متصل است و انتظار اجرای چنین کدی را ندارد. این نوع حمله برنامه‌های سیستم کاربر را مورد هدف قرار می‌دهد و نه خودِ برنامه‌ی شما را. اما از برنامه شما به‌عنوان یک وسیله برای حمله استفاده می‌کند. با توجه به اینکه اسکریپت روی مروگر کاربر از طرف یک سایت قابل اعتماد دانلود شده مرورگر راهی برای تشخیص غیرقانونی بودن آن اسکریپت ندارد و به ناچار آن را اجرا می‌کند. حتی تنظیمات امنیتی مرورگرِ Internet Explorer(منظور تنظیمات Security zones است) راهی برای مقابله ارائه نکرده است. معمولاً هدف این حملات دسترسی به کوکی‌های کاربران است. برای اجرای این حمله باید از کاربر خواسته شود تا برروی یک لینک که از نگاه او معتبر به نظر می‌رسدکلیک کند. با کلیک روی آن لینک مقدار پارامتر به جای یک رشته‌ی معتبر یک قطعه کد جاوا اسکریپت اعلام می‌شود. اگر برنامه‌ی سایت، محتوای رشته را بررسی نکرده باشد بطور مستقیم ویا غیرمسقیم دقیقاً قطعه کد تزریق شده را اجرا می‌کند.
- مثالی از یک لینک معتبر:

```
www.yourwebapplication.com/logon.aspx?username=bob
```

- همان لینک بصورت دستکاری شده و غیرمعتبر:

```
www.yourwebapplication.com/logon.aspx?username=<script>alert('hacker
code')</script>
```

مثال بالا یک پیغام pop-up که البته غیر مخرب است را در مرورگر کاربر نمایش می‌دهد. با استفاده از این حمله نفوذگر می‌تواند کوکی‌های کاربر را باز کرده و محتوای آن به سایت خودش پست کند و از این طریق نام کاربری و رمز عبور کاربر را سرقت کند.

- **اقدامات متقابل در برابر این حمله:**
- ترجیحا همه‌ی فیلدهای ورودی فرم‌ها را اعتبارسنجی کنید.
- همه‌ی ورودی‌های کاربران را یک نوع تهدید در نظربگیرید.
- ازعبارات منظم regular expressions برای اعتبارسنجی رشته‌های مختلف استفاده کنید.
- از توابعی که برای رمزنگاری URLها و مقادیر وارد شده‌ی پارامترهای URL کاربرد دارد استفاده کنید.
- این توابع اسکریپت‌های اجرایی را به کدهای مقابل آن‌ها در HTML که غیرمخرب است تبدیل می‌کند.

- توابع URL ecode و URL decode و html intities برای رمزنگاری رشته‌های URL بکار می‌روند.
- مثال: در قطعه کد زیر اگر به جای $foo عبارت:

```
<script>alert('hacker code')</script>
```

آیا قطعه کد اجرا خواهدشد؟

```
<?php
$foo=$_GET['Foo'];
$query_string='foo'.urlencode($foo);
Wcho $query_string;
?>
```

اگر تابع url ecode وجود نمی‌داشت Script اجرا می‌شد. اما اگراین تابع وجود داشت Script اجرا نمی‌شد.

- **SQL Injection**: در این حمله نفوذگر از مشکلات اعتبارسنجیِ ورودی‌ها سوءاستفاده می‌کند و دستورات دلخواهِ خود را برروی پایگاه داده‌ی سایت اجرا می‌کند. این حمله موقعی برروی برنامه‌ی شما اتفاق می‌افتد که دستورات SQL را بصورت پویا (یعنی با توجه به ورودی‌های کاربر) ساخته و به SQL Server ارسال کنید. هم‌چنین ممکن است در شرایطی که شما از string های ذخیره شده در سیستم کاربر بعنوان ورودی پایگاه داده استفاده کرده باشید و بر روی آن‌ها عملیات فیلترینگ انجام نداده باشید، رخ دهد این حمله بویژه در مواقعی رخ می‌دهد که برای اتصال به دیتابیس از یک اکانت با دسترسی بالا استفاده کرده باشید. این حمله زمانی اتفاق می‌افتد که یک ورودی از کاربر گرفته شده و بدون هر فیلترینگ مستقیماً در قالب یک query string قرار گیرد. بطور مثال از قطعه کد زیر را در نظربگیرید:

```
sqlDataAdapter myCommand = new SqlDataAdapter(
          "SELECT * FROM Users
          WHERE UserName='" + txtuid.textuid.text+ "'", conn);
```

در این قطعه کد مقدار وارد شده در فیلد textuid مستقیماً به دیتابیس ارسال شده است. اگر نفوذگر در این فیلد به جای نام کاربری عبارت زیر را وارد کند.

```
DROP TABLE Customers --
```

Query که نهایتاً برروی دیتابیس اجرا خواهد شد، بصورت زیر خواهد بود.

```
SELECT * FROM Users  WHERE UserName=' ';
```

با اجرای این دستور علاوه براینکه مشخصات یک کاربر از جدول user خوانده می‌شود، جدول customers بطور کامل مُنهدم می‌شود. (- - در انتهای دستور به SQL Server می‌فهماند که عبارات بعداز - - رادر نظر نگیریم، بطورمثال ممکن است بعد از نام کاربری شرط‌های دیگر نیز بیان شده باشد که با - در نظرگرفته نمی‌شود.) و یا یک نفوذگر می‌تواند در فیلد txtuid عبارت زیر را وارد کند:

```
' OR 1=1 -
```

که در اینصورت query string به شکل زیر خواهد بود:

```
SELECT * FROM Users  WHERE UserName ='' OR 1=1 -
```

چون ۱=۱ همیشه True است، هکر در حقیقت همه‌ی سطرهای جدول Users را اخذ می‌کند.

تمرین: عملیات Sql Injection را با Php پیاده‌سازی کنید.

- **اقدامات متقابل در برابر این حمله:**
- داده‌هایی که از طُرق مختلف به سیستم وارد می‌شوند، قبل از ارسال به دیتابیس بطورکامل اعتبارسنجی کنید، تا حاوی کارکترهایی مثل سینگل کوتیشن، دابل کوتیشن، بَک کوت و کلماتی مانند UNION و DROP و SELECT و... باشد.
- از تکنیک‌های مختلف برای خنثی کردن رشته‌های مخرب استفاده کنید. بطور مثال: استفاده از توابع Sprintf ویا ماژول PDO در PHP.
- مثال: قطعه کد زیر از ورود هر کارکتری بجز کارکترهای عددی به uid جلوگیری می‌کند:

```
$ uid  = $  -  GET['uid'];
$  uid = sprintf ("%d", $uid);
```

- در صورت امکان از اکانت‌هایی با دسترسی محدود برای اتصال به دیتابیس استفاده کنید. بطورمثال: اگر در بخش کاربریِ برنامه‌تان، فقط روال خواندن از دیتابیس لازم است، با کاربری که دسترسی update یا insert دارد به دیتابیس لاگین نکنید.
- **Canonicalizatin[1]**: در این نوع حمله نام یک فایل غیرمجاز طوری بعنوان ورودی به برنامه تزریق می‌شود که برنامه آن را یک نام مجاز تصور می‌کند. بنابراین این نوع حمله تأکیدش برروی نام منابع است، فایل‌ها، مسیرها و URLها نقاط آسیب‌پذیر برای این نوع حمله به حساب می‌آیند. بطور مثال قطعه کد زیر نام قالب را بطور مستقیم از کوکی‌های سیستم کاربر خوانده وتنظیمات قالب را لود می‌کند.

```
<?php
```

[1] قانونی سازی

```
$theme = $COOKIE[‘theme’];
Include (“themes/$theme/config.php”);
```

اگر نفوذگر در کوکی Theme وارد کند ” . . “ ، به جای اینکه فایل confing قالب include شود، فایل confingروتِ سایت incloude خواهد شد یعنی خواهیم داشت:

```
Themes/../config.php
```

که معادل قطعه کد زیر است:

```
/config.php
```

اما اگر قطعه کد بالا به‌صورت زیر نوشته شود ودر کوکیِ کاربر به جای نام قالب یک عدد که نماینده‌ی قالب است درج شود مشکل حل خواهد شد.

```
<?php
$theme = sprintf(“%d”,$_COOKIE[‘theme’]);
$themes = array(‘1’=> ‘default’,’2’=>‘blue’);
$theme=$themes[$theme];
Include (“themes/$theme/config.php”);
?>
```

این کد چطور از هک جلوگیری می‌کند؟

خط اول بررسی می‌کند که کوکی حتما عدد باشد و آن عدد به یک آرایه بنام theme داده می‌شود که این آرایه توسط کاربر و هکر قابل تغییر نیست. بنابراین دقیقاً نام قالبی از آن استخراج می‌شود که ما انتظار داریم یعنی defult یا bule ونه رشته مدنظر هکر.

- **اقدامات متقابل در برابر این حمله:**
- تا حد ممکن نام فایل را از کاربر دریافت نکنید و در عوض روشی بکار بگیرید که از طریق آن روش، پی به نامِ فایلِ موردنظر ببرید. بطور مثال به جای ذخیره کردن نام قالب در کوکی کد(id) آن قالب را ذخیره کنید و از طریق آن id نام قالب را بدست‌آورید.
- نام و مسیر فایل وارد شده را (اگر آن را از کاربر بعنوان ورودی دریافت می‌کنید) اعتبارسنجی کنید. برای مثال بررسی شود که مسیر واردشده در سلسله مراتب مسیرهای مجاز قرار داشته باشد یعنی مثلاً فایلی خارج از سطح www فراخوانی نشده باشد ویا مثلاً آدرس وارد شده حاوی /. . نباشد.

- مطمئن شوید که کارکترها واردشده از لحاظ incoding و طول نام و... با تنظیمات مدنظر شما هماهنگ باشد. بطورمثال وجود "- -" در نام یا مسیر فایل می‌تواند حاکی از حمله‌ی SQL Injection باشد.

۵-۱-۲ نکات مربوط به اهرازهویت در برنامه طراحی شده توسط شما: نسبت به نیازمندی‌های شما مکانزیم‌های احراز هویتی مختلفی وجود دارند که باید از بین آن‌ها یک یا چندتا را انتخاب کنید. اگر در انتخاب یا پیاده‌سازی آن مکانیزم درست عمل نکنید تهدیدهایی زیربرنامه‌ی شما را به خطر خواهد انداخت.

- **Network Eavesdropping[1]:** اگر credential های احرازهویت بصورت متن‌ساده[2] بین کلاینت و سرور رد و بدل شود، نفوذگری که به یکی از نرم‌افزارهای مانیتور شبکه مانند Wire shark مجهز است، می‌تواند ترافیک شبکه را کپچر کرده و یوزرنیم و پسوردهای رد و بدل شده را بدست آورد.

- **اقدامات متقابل در برابر این تهدید:**
- از مکانزیم‌های احراز هویتی استفاده کنید که لازم نباشد پسورد از طریق شبکه ارسال شود مکانیزم‌هایی مانند: پروتکل kerberos و یا windows authentication.
- اگر لازم است پسوردها از طریق شبکه منتقل شوند، مطمئن شوید که به‌صورت رمزنگاری شده منقل می‌شوند. بطورمثال قبل از ارسال پسورد از طریق کاربر به سرور می‌توان آن را با ScriptJava هش کرده و سپس ارسال کرد و یا کاناهای ارتباطی رمزنگاری شده مانند SSL استفاده کنید.

در مورد پروتکل Kerberos تحقیق کنید.

- **حملات Brute force:** حملات Brute force مبتنی برمحاسباتی هستند که منجر به شکستن پسوردهای هش شده و یا داده‌های Encrypt شده می‌شود و برای مقابله با این تهدید، تنها راه، استفاده از پسوردهای قوی[3] است.

---

[1] نشتِ اطلاعات از طریق شبکه
[2] plaintext
[3] Strong

- **حملات Dictionary[1]:** این حمله برای بدست‌آوردن پسوردها استفاده می‌شود در اکثر سیستم‌ها پسوردها به صورت هش شده ذخیره می‌شوند.(دقت کنید که پسوردهای نمی‌توانند بصورت encrypt شده ذخیره شوند، چرا که در صورت از دست دادن کلید رمزنگاری تمام پسوردها غیر قابل بازگشت به حالت اول خواهد شد.) حمله کننده در این نوع حمله یک دیکشنری از تمامی لغاتی که حدس می‌زند کاربران بعنوان پسورد انتخاب می‌کنند(مانند: admin123 یا ۱۲۳۴۵۶ و...) بدست می‌آورد و به سادگی تمامی آن‌ها را هش کرده و در مقابل آن‌ها، در دیکشنری ذخیره می‌کند. در صورتیکه پسورد هش‌شده‌ی یک از کاربران را بدست‌آورد براحتی به‌راحتی می‌تواند پسورد اصلی را از داخل این دیکشنری با یک Query ساده بدست‌آورد. سایت‌هایی مانند : Crackstation.net و md5cracker.org برای شکستن پسوردها ی هش شده کاربرد دارند.

- **اقدامات متقابل در برابراین نوع حمله:**
- از پسوردهای پیچده‌ای که از یک روال قابل پیش‌بینی برای تولید آن‌ها استفاده کرده باشید، استفاده نکنید. بطورمثال : چهاررقم آخرموبایل یا تاریخ تولد یکی از اشخاص خانواده ، تاریخ ازدواج و... ترکیبی از حروف کوچک، بزرگ، عدد و کارکترهای غیرالفبایی استفاده کنید و کاربران را مجبور به انتخاب چنین پسوردهایی کنید.
- پسوردها را در سمت کاربر هرگز بصورت Reversible[2] ذخیره نکنید.

- **حملات Cookie Replay :** در این نوع حمله، حمله کننده کوکی کاربران و یا محتوای کوکی آن‌ها را به روش‌های مختلف (مانیتورکردن شبکه و یا حضورفیزیکی در پشت سیستم کاربر و...) سرقت کرده و برروی سیستم خود کوکی‌های با آن محتوا ایجاد و در نتیجه به حساب کاربری کاربر دسترسی پیدا می‌کند.

تمرین عملی: در یک سایت که از کوکی برای احراز هویت استفاده می‌کند لاگین کنید و با سرقت کوکی‌های ایجاد شده، در سیستم دیگری از کوکی‌های سرقت شده استفاده کرده و بدون داشتن نام کاربری و پسود لاگین شوید (برای این کار به افزونه‌ی cookie manager نیاز است)

- **اقدامات متقابل در برابر این نوع حمله:**
- از کانالهای ارتباطی رمزنگاری شده با SSL استفاده کنید.

---

[1] حملات لغت‌نامه

[2] قابل برگشت

- یک Time out مشخص برای کوکی یعنی یک لحظه مرگ مشخص برای کوکی‌ها در نظر بگیرید. و به کاربران اجازه دهید متناسب با محلی که از آنجا به سیستم لاگین می‌کنند عمرکوکی را انتخاب کنند. بطورمثال بتوانند انتخاب کنند که کوکی یک روزیا یک هفته یا یک ماه برروی سیستم‌شان بنشیند.
- ترفندهای غیر رسمی برای مقابله با این تهدید وجود دارد، از جمله: نام کوکی‌ها را گیج کننده انتخاب کنید و یا یک سری کوکی که کاربرد خاصی ندارد و فقط حکمِ گمراه کردنِ هکر را دارند ایجاد کنید. هم‌چنین محتوای کوکی‌ها را تاحدممکن پیچیده و گمراه کننده در نظربگیرید.

۳-۱-۵ **نکات مربوط به Authorization[1] در برنامه طراحی شده توسط شما:** بر مبنای Id کاربر و نقش او در محیط عملیاتی، سطح دسترسی او به یک منبع یا سرویس خاص تعیین می‌شود. اگر این عملیاتِ تعیین مسئولیت، به درستی انجام نشود تهدیدات زیر برنامه را به خطر خواهد انداخت:

- **Elevation of privilege[2]:** اگر حمله کننده بتواند با عملیاتی مانند عضوکردن حساب کاربری خود در گروه مدیران سطح دسترسی خودرا بالا ببرد. می‌تواند به منابعی که مدیران دسترسی دارند، دسترسی داشته باشد.
- **راه اصلی مقابله با این تهدید:**
- حداقل سطح دسترسی‌ها را به processها و user accountها و Serviceها بدهیم.
- لاگ انداختن فعالیت کاربران در سیستم ویا پیاده‌سازی یکIds در برنامه است. که بطورمثال اگریک کاربری که تا پیش از این به پنل مدیریت سیستم دسترسی نداشته، برای اولین بار در آن پنل مشاهده می‌شود احتمال یک تهدید بدهیم و نام کاربری او را به مدیر کل سیستم گزارش کنیم.
- **Disclosure of confidential data[3]:** زمانی رخ می‌دهد که کاربران غیرمجاز[4] به اطلاعات حساس[5] دسترسی پیدا کنند. بطورمثال: اطلاعاتی مانند شماره‌ی کارت اعتباری، جزئیات خصوصیات کارمندان، رکوردهای مالی و امثال آن در سطح کاربر و اطلاعاتی مانند اطلاعات احراز

---

[1] احراز مسئولیت
[2] بالابردن سطح دسترسی
[3] افشای اطلاعات محرمانه
[4] unauthorized users
[5] sensitive data

هویتِ اکانت‌های خدماتی و database connection string[1] در سطح پیکربندی برنامه، نمونه‌هایی از افشای اطلاعات هستند.

- **اقدامات متقابل در برابر این تهدید:**
- قبل از دسترسی دادن برای انجام یک عملیات که آن عملیات می‌تواند بطور بالقوه به داده‌های حساس دسترسی بدهد سطح دسترسی کاربر را بررسی کنید. بطورمثال قطعه کد زیر که قبل از درج داده در دیتابیس اجرا می‌شود سطح دسترسی مدیر را برای درج در دیتابیس بررسی می‌کند. role[2]

```
//Check Permission
if($admin->AdminPermission($ausername,"allow_admins_add"))
{
    $counts = $admin->AddAdmin($acode,md5($apass),$aactive,$afname,$alname,$
}
else
{
    $counts = 0;
    $error = _ADMIN_YOU_DO_NOT_HAVE_NECCESSARY_PERMISSIONS.'.<br>';
}
if($counts==1)
{
    //Rename Uploaded Image
    if(!empty($apic))
```

شکل5 -1 بررسی سطح دسترسی مدیر

- سطح دسترسی منابع مختلف برنامه را در سطح هاست بررسی کنید. بطورمثال لزومی ندارد config.php که حاوی اطلاعات حساس به دیتابیس است سطح دسترسی‌اش 777 باشد. (همان‌طور که می‌دانید سطح دسترسی پیش‌فرض‌های فایل‌ها در لینوکس ۶۴۴ و سطح دسترسی پیش‌فرض پوشه‌ها ۷۵۵ باید باشد.)
- از استانداردهای رمزنگاری برای ذخیره کردن داده‌های حساس در فایل‌های config و دیتابیس استفاده کنید.

Data Tampering[3]: این تهدید به ویرایش غیرمجاز داده‌ها اشاره دارد.

---

[1] رشته‌ی اتصال به دیتابیس

[2] نقش

[3] دستکاری داده‌ها

- **اقدامات متقابل در برابر این نوع تهدید:**
- از کنترل‌های دسترسی سخت‌گیرانه برای اینکه مطمئن شوید فقط کاربران مجاز به داده‌ها دسترسی دارند و می‌توانند آن‌ها را ویرایش کنند، استفاده کنید.
- از امنیت مبتنی بر نقش[1] برای تمایز بین کاربرانی که می‌توانند داده‌ها را ببینند یا آن‌ها را ویرایش کنند، استفاده کنید. (بطورمثال در برخی CMS ها مانند وردپرس برای کاربران نقش‌هایی مانند: مشترک ، نویسنده، مدیر، مدیر کل و... در نظرگرفته شده است.)
- **حملات Luring[2]:** این نوع حمله زمانی رخ می‌دهد که یک موجودیت با سطح دسترسی کم قادر است دسترسی یک موجودیت با سطح دسترسی بیشتر را بدست‌آورده و از طرف او عملیاتی را اتجام دهد.

- **اقدامات متقابل دربرابراین تهدید:**
- دسترسی به کدهای حساس را محدود به احراز هویت‌های سخت‌گیرانه‌تر کنید.

4-1-5 نکات مربوط به Configuration Management[3] در طراحی برنامه: بسیاری از برنامه‌ها از interface هایی برای مدیریت، تنظیمات و اجازه به مدیران جهت تغییر پارامترهای مربوط به تنظیمات و آپدیت محتوای صفحات وب و عملیات مربوط به نگه‌داری سیستم استفاده می‌کند. مهم‌ترین تهدیدات در این زمینه عبارتند از:

- **دسترسی غیرمجاز به واسطهای مدیریتی:** اکثر برنامه‌ها یک پنل مدیریتی تحت وب دارند که به administrators و operators و توسعه‌دهندگانِ محتوا، اجازه‌ی مدیریت محتوا و پیکربندی سایت را می‌دهد. قطعاً دسترسی کاربران غیر مجاز به این پنل تهدیداتی رادر پیش دارد.

- **اقدامات متقابل در برابر این نوع تهدید :**
- تاحد ممکن تعداد واسطهای مدیریتی را کاهش دهید.
- از احرازهویت سخت‌گیرانه استفاده کنید.

[1] role- based security
[2] فریفتن
[3] مدیریت پیکربندی‌ها

- در شبکه‌های تحت دامین ممکن است بخواهید فقط دسترسی به مدیرانِ محلی[1] بدهید. و اگر حتی لازم شد به کاربرانِ remote دسترسی داده شود، از VPNهایی که با تکنولوژی SSL رمزنگاری شده باشند استفاده کنید و یا سیاست IPsec را فعال کنید.

- **دسترسی غیر مجاز به فایل‌ها و مخازن تنظیمات و پیکربندی:**

- **اقدامات متقابل در برابر این تهدید:**
- سطح دسترسی فایل‌های پیکربندیِ متنیِ حساس مانند: Machine.config و web.config و config .php، محدود کنید.
- برخی از مخازن پیکربندی را خارج از فضای وب نگه‌دارید. این نکته امکان دانلود این فایل‌ها و ایجاد تهدید را کاهش‌می‌دهد. بطور مثال می‌توانید فایل concfig.php را در پوشه‌ی etc که خارج از سطح www است نگه دارید.
- **بازیابی کلیدهای رمزنگاری ذخیره شده در فایل‌های پیکربندی متنی ساده:** ممکن است پسوردها و connection stringها را رمزنگاری کرده باشید و کلید رمزنگاری را در یک فایل متنی ذخیره کرده باشید و این فایل متنی سطح دسترسی مناسبی نداشته باشد.

- **Lack of Individual Accountability[2] :**
- باید برای هر مدیر یک اکانت مخصوص به او ساخته شود.
- اکانت‌های مدیریتی نباید بین چند نفر به اشتراک گذاشته شود.
- اکانت‌های مربوط به user یا application یا service از هم متمایز شود، و سطح دسترسی هریک بطور جداگانه تعیین شود.

- **Over privilegd Application and service Accounts[3]:** برنامه برای انجام کارهای مختلف نیاز به دسترسی به سرویس‌هایی دارد مانند: سرویس دیتابیس که این سرویس‌ها نیاز به یک اکانت مخصوص به خود دارند (.مانند اکانتی که برای اتصال به دیتابیس ساخته می‌شود) علاوه براینکه برای کارهای مختلف اکانت‌های مختلفی تولید می‌کنید سیاستی تعیین کنید که هر اکانت حداقل دسترسی را به آن سرویس داشته باشد.

---

[1] Local administrator

[2] عدم وجود اکانت برای هر شخص با مسئولیت مختص به خود

[3] اکانت‌های مربوط به اپلیکیشن وسرویس که بیش از نیاز دسترسی داشته باشند

۵-۱-۵ تهدیدات مربوط به دسترسی به داده‌های حساس در طراحی برنامه:

- دسترسی به اطلاعات حساس ذخیره شده در مخازنی مانند دیتابیس و فایل‌ها.
- نشت اطلاعات از طریق شبکه
- دستکاری داده ها

۵-۱-۶ تهدیدات مربوط به مدیریت جلسه در طراحی برنامه:

- سرقت سشن
- استفاده از سشن یک کاربر معتبر
- حملات مردی در میان

۵-۱-۷ تهدیدات مربوط به رمزنگاری در طراحی برنامه:

- تولید و مدیریت کلید رمزنگاری با الگوریتم‌های ضعیف.
- رمزنگاری ضعیف یا بدون قاعده و مختص به خود(من درآوردی) (حدس روش و کلید هش،

گاهی اوقات هکر با بررسی کدهش شده، الگوریتم هش که توسط برنامه‌نویس استفاده شده است را حدس می‌زند.)

۵-۱-۸ تهدیدات مربوط به دستکاری پارامترها در طراحی برنامه:

- **دستکاری Query string** : به پارامترها و مقدار آن‌ها که به روش GET یا POST به سرور ارسال می‌شود در اصطلاح Query string گفته می‌شود. که هکر می‌تواند با دستکاری این رشته به اهداف خود برسد.
- **دستکاری فیلدهای فرم**
- **دستکاری کوکی‌ها**
- **دستکاری هدر http**

۵-۱-۹ تهدیدات مربوط به مدیریت استثناها و خطاهای برنامه:

- **Information disclosure**: گاهی اوقات هکر از روی خطاهایی که برنامه نمایش می‌دهد می‌تواند اطلاعات مفیدی بدست آورد و برای نفوذ از آن‌ها استفاده کند. بعنوان مثال هکر از روی خطاها به جزئیات زیرساخت برنامه یا هاست یا شبکه پی می‌برد. بطورمثال هکر از روی خطاها به

جزئیات زیرساخت برنامه یا هاست پی می‌برد. و همچنین پی می‌برد که برنامه نویس از چه پایگاه داده‌ای استفاده می‌کند.

- **اقدامات متقابل در برابر این نوع تهدید:**

- از روش‌های برخورد با استثناها[1] در کدنویسی استفاده کنید.
- ردپای استثناها و خطاهایی که در سیستم رخ می‌دهد را ثبت کنید[2].
- در برابر خطاهای رخداده پیغام‌های خطای بی‌ضرر و کلی به کلاینت نمایش دهید.

- **Denial of service**: هکر از اطلاعات بدست‌آمده در قالب حملات DOS استفاده می‌کند، اگر هکر بتواند از روی خطاهایی که سیستم نمایش می‌دهد. به این نتیجه برسد که سیستم در برابر حملاتDOS مقاوم نیست می‌تواند از نقاط آسیب‌پذیر برای این نوع حمله استفاده کند، بطورمثال پس از ارسال یک ورودی به سیستم خطای Time out نشان می‌دهد. این نشان می‌دهد که سیستم زمان زیادی را صرف پردازش آن ورودی کرده است.

۱۰-۱-۵ تهدیدات مربوط به زیرنظرگرفتن و ثبت ردپا: اگر ردپای کامل و دقیقی در سیستم ثبت نشود تهدیدات زیر، سیستم را تهدید خواهد کرد:

- کاربر انجام یک عملیات را انکار می‌کند.
- حمله کننده به یک برنامه نفوذ می‌کند بدون اینکه هیچ نوع ردپایی از خود به جا بگذارد.
- حمله کننده ردپای خود را مخفی می‌کند.

لغات و اصطلاحات مهم در زمینه‌ی برنامه‌های مخرب :

- **Virus** : بدافزاری که وقتی اجرا می‌شود سعی می‌کند خودش را برروی کدهای قابل اجرای دیگر تکثیر کند، وقتی این عملیات با موفقیت انجام شود، در اصطلاح گفته می‌شود آن کد آلوده[3] شده است. وقتی آن کدِ آلوده اجرا شده است، در حقیقت ویروس نیز اجرا شده است.
- **Worm** : یک برنامه‌ی کامپیوتری که می‌تواند بطور مستقل اجرا شود و یک نسخه‌ی کامل از خود را به هاست‌های دیگرِ یک شبکه منتشر کند.
- **Logicbomb[4]**: Program که از طریق یک مزاحم[1] به داخل یک Software تزریق شده است. یک Logicbomb تا زمانی‌که شرایط لازم تأمین نشود به‌صورت خفته[2] وجود دارد اما به محض اینکه پیش‌نیازها فراهم شد یک عملیات غیرمجاز را انجام می‌دهد.

---

[1] Exception handeling
[2] Logging
[3] infected
[4] بمب منطقی

- **Trojan horse:** یک برنامه کامپیوتری که به نظر می‌رسد یک کارمفید انجام می‌دهد اما در واقع یک عملیات مخرب بالقوه و مخفی در بطن خود دارد که مکانیزم‌های امنیتی را دور می‌زند.
- **Back door[3]:** هرمکانیزمی که وارسی‌های امنیتی معمول را دور بزند[4] و یک دسترسی غیرمجاز به یک عملکرد بدهد.
- **Mobile code[5]**: یک نرم افزار مانند Script و macro و دیگر دستورالعمل‌های قابل حمل که می‌تواند به‌صورت دستکاری نشده، ناهمگون به بسته‌های اجرایی مختلف، که لزوماً یک نوع نیستند(مثلا: به زبان‌های برنامه نویسی مختلف یا سیستم‌عامل‌های مختلف)، منتقل شده و به شکل موردنظر هکرانجام شود.
- **Exploits[6]:** کدخاصی که برای یک آسیب‌پذیری خاص، و یا یک مجموعه از آسیب‌پذیری‌ها نوشته شده است.
- **Downloaders[7]**: برنامه‌ای که آیتم‌های دیگری که هکر نیاز دارد روی ماشین تحت حمله نصب می‌کند. Downloaders معمولا از طریق ایمیل به قربانی ارسال می‌شود.
- **Auto-rooter:** یک ابزار مخرب برای هکر، که از طریق آن به یک ماشین جدید بصورت remote[8] نفوذکند.
- **Kit(virus generator):** یک مجموعه ابزار برای تولید خودکار ویروس‌های جدید.

تمرین: یک عدد Kit پیدا کنید.

- **Spammer programs :** برنامه‌ی کدی که برای ارسال حجم عظیمی از ایمیل‌های ناخواسته (معمولاً تبلیغاتی) استفاده می‌شود.
- **Flooders :** برنامه‌ی کدی که برای حمله به سیستم‌های کامپیوتری شبکه شده استفاده می‌شود، به این صورت که حجم عظیمی از داده‌های غیرمفید به این سیستم‌ها ردوبدل می‌کند و از این طریق ترافیک شبکه را اشغال کرده و یک نوع حمله DOS صورت می‌دهد.
- **Keyloggers**: کلیدهای فشرده شده بر روی کیبورد در سیستم قربانی را کپچرکرده و برای هکر می‌فرستد.

---

[1] intruder
[2] dorment
[3] در پشتی
[4] bypass
[5] کد محرک(سیار)
[6] سوءاستفاده
[7] Downloader
[8] از راه دور

- **Rootkit:** از مجموعه ابزارهایی که هکر بعد از شکستن کامپیوتر قربانی و دسترسی به آن، در سطح root از آن‌ها استفاده می‌کند.
- **Zombie,bot :** برنامه‌ای که روی یک ماشین آلوده که برای حمله به ماشین های دیگر مورد استفاده قرارمی‌گیرد فعال شده است.

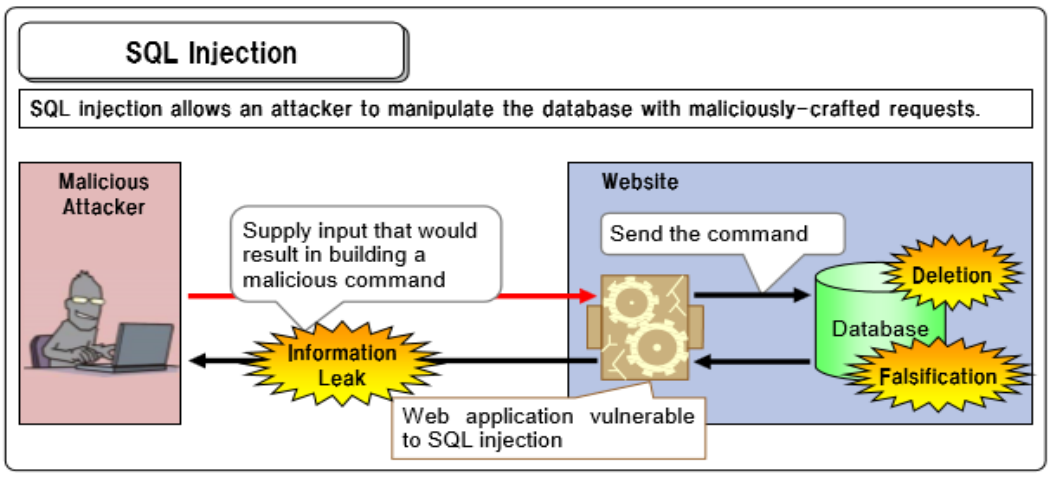
  تحقیق: در مورد فلسفه کلمه زامبی تحقیق کنید.
- **Spyware** : نرم‌افزاری که اطلاعاتی از کامپیوتر جمع‌آوری می‌کند وآن‌ها را به یک سیستم دیگر می‌فرستد.
- **Adware** : تبلیغاتی که در یک نرم‌افزار تزریق شده است و می‌تواند باعث ایجاد pop-upهای تبلیغاتی و یا انتقال مرورگر به یک سایت تجاری که موردنظر هکر است، شود.

فصل ششم

۶-۱ مهم‌ترین تهدیدات برنامه‌های تحت وب:

۶-۱-۱ SQL Injection[1]
۶-۱-۲ OS command injection[2]
۶-۱-۳ Unchecked path parameter/Distory traversal
۶-۱-۴ Improrer session management
۶-۱-۵ cross-site scripting
۶-۱-۶ CSRF cross-site request forgery
۶-۱-۷ HTTP header injection
۶-۱-۸ Mail header injection
۶-۱-۹ lack of authentication and authorization

1-1-6 SQL Injection[3]: اکثر برنامه‌های تحت وب دستورات SQL خود براساس ورودی‌هایی که از کاربر میگیرند می‌سازند (مثلاً کاربر عبارتی را جستجو می‌کند و این عبارت در ساخت یک کوئری select قرار می‌گیرد. بنابراین اگر فرآیند ساخت دستور SQL امن نباشد. حمله به و دستکاری دیتابیس ممکن خواهد بود. این مشکل در اصطلاح تهدید تزریق[4] SQL نامیده می‌شود و حمله‌ای که از طریق سوءاستفاده از این تهدید استفاده می‌شود، حمله‌ی تزریقِ "SQL " نامیده می‌شود. شکل کلی این نوع حمله به صورت زیر است:



[1] تزریق کدِ اس کیو ال
[2] تزریق دستورات سیستم‌عاملی
[3] تزریق
[4] Sql injection vulnerabitily

شکل ۶-۱ شمای کلی حمله SQL injection

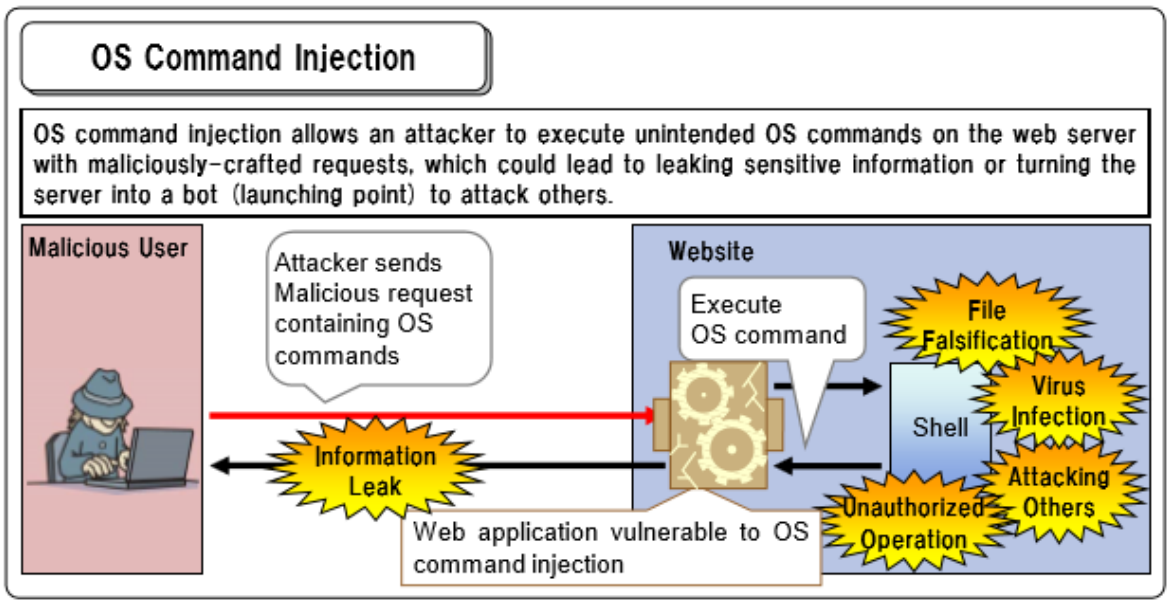
- **تهدیدات احتمالی این مشکل:**

- دسترسی به داده‌های حساس ذخیره شده در دیتابیس. بطورمثال: کشف اطلاعات شخصی کاربران .
- تغییر یا حذف داده‌های ذخیره شده در دیتابیس. بطورمثال: تغییر محتوای صفحات وب، تغییر پسوردها، خاموش کردن یک وب‌سرویس.
- دور زدن اهراز هویت هنگام لاگین. بطورمثال بدون داشتن نام کاربری و رمز مدیر کارهایی که در پنل مدیریت انجام می‌شود را می‌توان انجام داد.
- اجرای فرمان های در سطح سیستم عامل با استفاده از proceduresهای ذخیره شده در دیتابیس. بطور مثال می‌توان از سیستم قربانی بعنوان یک bot یا zombie برای حمله به دیگران استفاده کرد و یا کنترل سیستم را در اختیار گرفت.

- وب‌سایت‌هایی نیاز به توجه ویژه دارند: هر وب‌سایتی که از برنامه‌ای استفاده می‌کند که با دیتابیس سر و کار دارد می‌تواند قربانی این حمله باشد.
- تهدیدات گزارش شده در این حمله: حدود ۱۴٪ تهدیدات مرتبط با وب‌سایت‌ها که به IPA[1] گزارش شده است . مربوط به تهدیدات تزریق SQL بوده است.
- راه‌های پایه‌ای :
  - تمام دستورات SQL را با استفاده از Place holderها در برنامه‌نویسی تولید کنید.
  - از دستورات موتور دیتابیس برای رفع مشکلات امنیتی احتمالی ورودی‌های کاربرکمک بگیرید.
  - دستورات SQL را بطورمستقیم از طرف کاربر به برنامه ارسال نکنید.
  - اطلاعاتی که هنگام بروز خطا به کاربر نمایش می‌دهید را محدود کنید .
  - حداقل دسترسی‌ها را به اکانت‌های پایگاه داده بدهید.

[1] مخزن اطلاعات مربوط به تهدیدات ژاپن

2-1-6 تزریق دستورات سیستم‌عاملی: برنامه‌های تحت وب می‌توانند تهدیدی ایجاد کنند که به هکر اجازه دهد دستورات در سطح سیستم‌عامل را از طریق آن برنامه اجرا کند.به این مشکل در اصطلاح " تزریق دستورات سیستم‌عاملی " و به حمله‌ای که از طریق این مشکل صورت می‌گیرد حمله‌ی دستور سیستم‌عاملی گفته می‌شود.



شکل ۶-۲ شمای کلی OS command injection

- تهدیدات احتمالی:

- هکر می‌تواند فایل‌های ذخیره شده روی سرور را ببیند را تغییر دهد یا حذف کند . بطورمثال:کشف اطلاعات حساس و یا تغییر فایل‌های config .
- هکر می‌تواند سیستم را بطور مخربی دستکاری کند. بطور مثال: سیستم‌عامل را خاموش کند useraccountهایی ایجاد یا حذف کند.
- هکر می‌تواند برنامه‌های مخربی روی سیستم بارگذاری و اجرا کند. بطورمثال: ویروس، کرم، bot، ایجاد Back door.
- هکر می‌تواند از سیستم به عنوان نقطه‌ای برای حمله به دیگران استفاده کند. بطور مثال: پیاده‌سازی حمله‌ی DOS از روی سرور قربانی روی سایت‌ها وسرورهای دیگران و یا ارسال Spam از طریق سیستم قربانی.

- وب‌سایت‌هایی که نیاز به توجه ویژه دارند: وب‌سایت‌هایی که از برنامه استفاده می‌کنند که آن برنامه‌ها از توابع در سطح سیستم‌عامل استفاده می‌کند و یا یک برنامه‌ی خارجی را فراخوانی می‌کند.

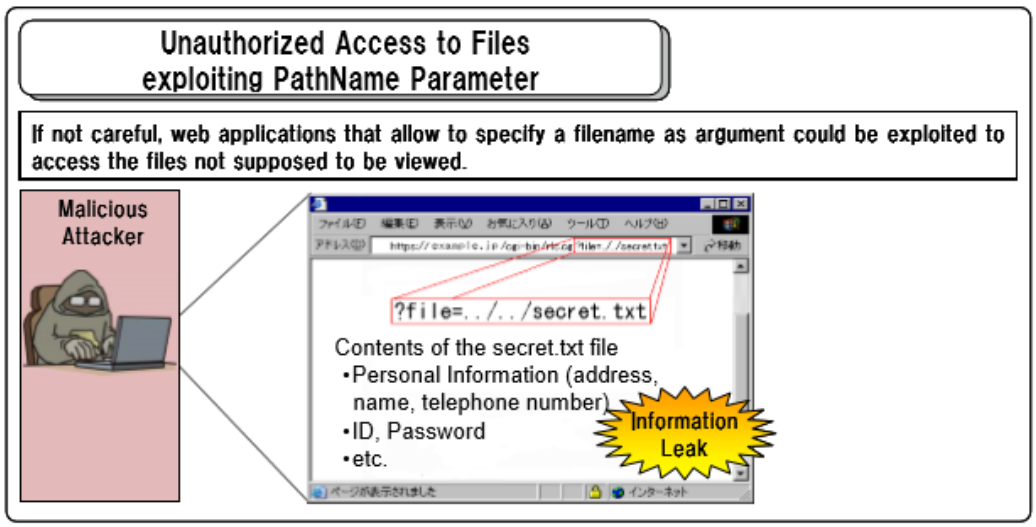
بطورمثال: توابع زیر در دو زبان برنامه‌نویسی Perl و Php می‌توانند نقاط آسیب‌پذیر باشند.

Perl	Php
Open() Eval() System()	Exec() Passthru() Open() System() Shell- Exec()

- تهدیدات گزارش شده در این حمله: طبق آنچه به IPA گزارش شده است اکثر برنامه‌های تحت وبی که به زبان perl نوشته شده‌اند، قربانی اصلی این نوع حمله هستند.
- راه‌حل‌های پایه‌ای :

- از استفاده از توابعی که فرمان‌های در سطح سیستم‌عامل را فراخوانی می‌کند، تا حد ممکن اجتناب کنید.
- اگر مجبور به استفاده از این توابع شدید پارامترهای ورودی آن‌ها را بطور دقیق بررسی کنید که مسیرهای غیرمجاز را فراخوانی نکند.

3-1-6 پارامترهای بررسی نشده مسیر یک فایل یا پیمایش پوشه‌ها: برخی برنامه‌های تحت وب اجازه می‌دهند که نام و مسیر یک فایلِ ذخیره شده روی سرور بطورمستقیم از طریق یک پارامترخارجی (بطور مثال پارامتری در نوار آدرس) مشخص شود. اگر اینگونه برنامه با دقت برنامه‌نویسی نشده باشند. هکر می‌تواند یک فایل دلخواه را آدرس‌دهی کند و برنامه آن فایل را اجرا کند . این مشکل در اصطلاح "تهدید پیمایش مسیر[1] " و به حمله‌ای که از طریق این تهدید اجرا شود حمله‌ی "پیمایش مسیر[2]" گفته می‌شود. شکل کلی این حمله بصورت زیر است:



شکل ۶-۳ شمای کلی حمله‌ی پیمایش مسیر

- تهدیدات احتمالی: هکر می‌تواند فایل‌های ذخیره شده روی سرور را ببیند و آن‌ها را ویرایش یا حذف کند.
- وب‌سایت‌هایی که نیاز به توجه ویژه دارند:
  - وب‌سایت‌هایی که از برنامه‌هایی استفاده می‌کنند که نام یا مسیر یک قایل را از طریق پارامترهای خارجی دریافت می‌کند. مثال: برخی برنامه‌ها نام ومسیر قالب سایت را از طریق یک پارامتر یا کوکی ذخیره شده روی سیستم کاربر می‌خواند.
  - برخی برنامه‌ها ورودی‌های کاربر را در فایل‌هایی ذخیره می‌کنند که کاربر مشخص می‌کند. (مانند بخش ویرایش فایل در وردپرس)
- تهدیدات گزارش شده : درصد کمی از وب‌سایت‌های هک شده مربوط به این حمله بوده‌اند اما آمار آن‌ها در حال افزایش است.
- راه‌حل‌های پایه‌ای :

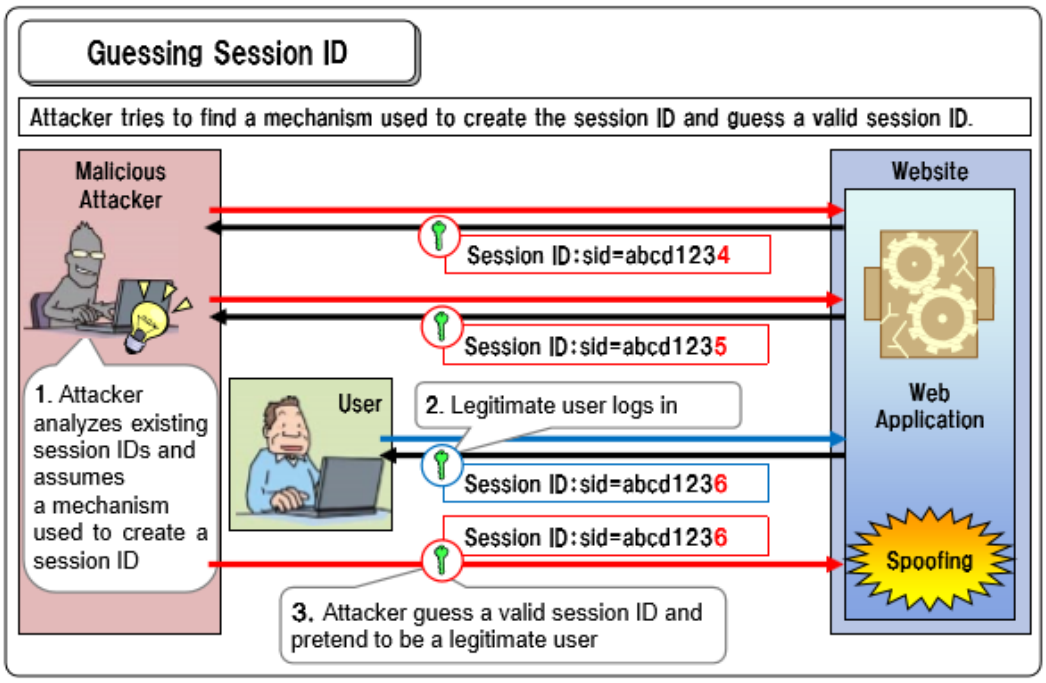
[1] Directory traversal vulnerability
[2] Directory traversal attack

- نام و یا مسیر فایل‌های ذخیره شده روی سرور را از پارامترهای خارجی بطور مستقیم دریافت نکنید.
- هنگام فراخوانی یک فایل از مسیرها فیکس شده (بدون استفاده از متغیر در بین مسیر) استفاده کنید.
- سطح دسترسی فایل‌ها را بطور صحیح مدیریت کنید.
- نام فایل‌ها را بررسی کنید تا فایل‌های غیرمجاز فراخوانی نشده باشد.

۴-۱-۶ **مدیریت نادرست سشن:** پیش از این در بحث session hijacking توضیح داده شد. برخی برنامه‌های تحت وب یک Seesion Id برای هر کاربر جهت شناسلیی او صادر می‌کند اگر این Seesion Id به‌درستی تولید و مدیریت نشود. هکر می‌تواند Seesion Id مربوط به یک کاربرقانونی را دزدیده و بطورغیرقانونی به خدمات مختص کاربران قانونی دسترسی پیدا کند. به حمله‌ای که از طریق این تهدید صورت می‌گیرد در اصطلاح سرقت جلسه[1] گفته می‌شود.

- حالت‌های مختلف این حمله:

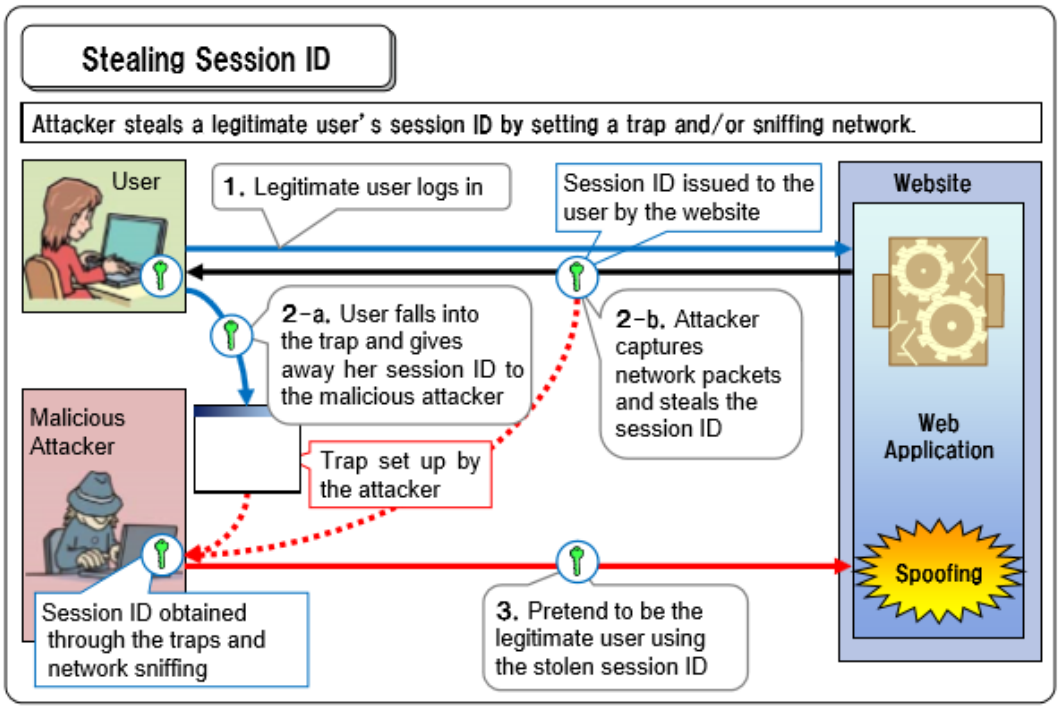
حدس Session Id به‌صورت شکل زیر است:



شکل ۶-۴ حدس سشن آی دی

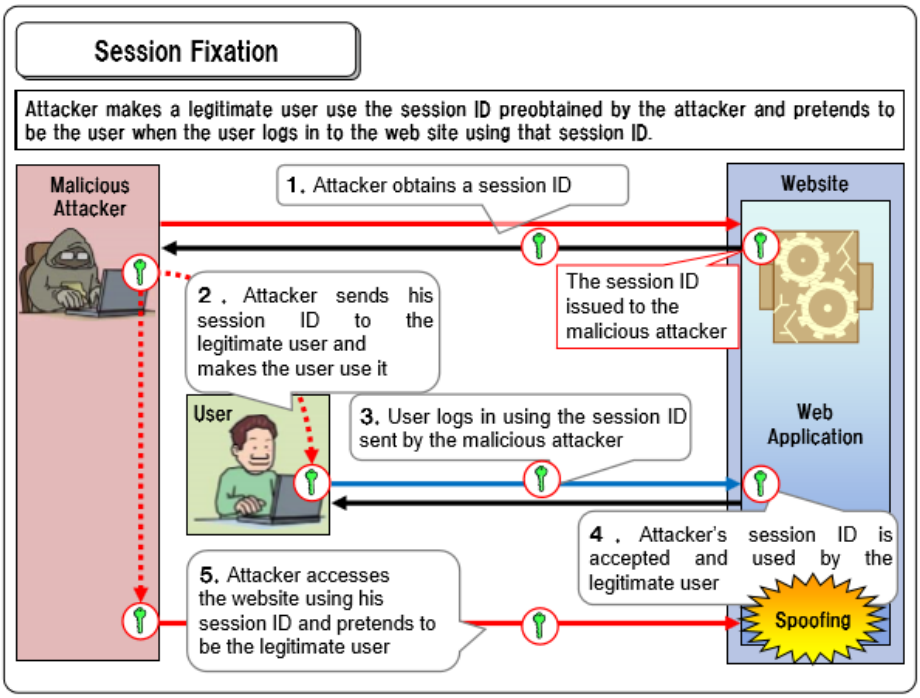
[1] Session Hjacking

سرقت Session Id بصورت شکل زیر است.



شکل ۶-۵ سرقت سشن آی دی

تحمیل Session Id: در این حالت هکر یک کاربرقانونی را مجبور می‌کند که از Session Id خودش (هکر) استفاده کند. به‌صورت شکل زیر:



شکل ۶-۶ تحمیل سشن آی دی

- تهدیدات احتمالی:

- دسترسی به خدماتی که در حالت عادی مختص کاربرانی است که بطور مجاز به سیستم لاگین کرده‌اند. بطورمثال: انتقال وجه از حساب یک کاربر، خرید کالا به حساب یک کاربر، لغو عضویت کاربر از یک سرویس و...
- افزودن و ویرایش اطلاعاتی که در حالت عادی فقط کاربران لاگین شده مجاز به دسترسی به آن‌ها هستند. بطورمثال: تغییر پسوردها، یا درج داده‌های اشتباه.

- وب‌سایت‌هایی نیاز به توجه ویژه دارند: وب‌سایت‌هایی که از برنامه استفاده می‌کنند که نیاز دارد که کاربر در آن‌ها لاگین کرده باشد تا از خدمات خاصی مانند پرداخت آنلاین استفاده کند.
- تهدیدات گزارش شده : فقط درصد کمی از وب‌سایت‌های گزارش شده به IPA از طریق این تهدید هک شده‌اند اما آمار آن‌ها در حال افزایش است.
- راه‌حل‌های پایه‌ای:

- حدس سشن آیدی را سخت کنید.
- Session Id را از طریق پارامترهای URL ارسال نکنید.
- در صورتی‌که از https استفاده می‌کنید تنظیمات امن کردن کوکی را فعال کنید، این تنظیمات باعث می‌شودکوکی به صورت رمزنگاری شده از طریق پروتکل https به سرور ارسال شود.
- بعد از لاگین کردن کاربر یک session Id جدید به او نسبت دهید.
- از session id رندوم استفاده کنید.
- اگر session id در کوکی ذخیره می‌کنید عمر کوکی را بادقت ذخیره کنید.

منابع

- کتاب Improving web Application Security (بهبود امنیت برنامه‌های تحت وب).
- Threats and Countermeasures (تهدیدها و اقدامات متقابل) از انتشارات شرکت مایکروسافت.

ابزارهای مفید برای حفظ امنیت وب‌سایت

مانند ابزار Event viewer و Laster visitors در ویندوز (آخرین بازدید کنندگان سایت را نمایش می‌دهد) و ابزار Webalizer وAnalog stats (جزئی‌ترین دسترسی‌ها به فایل‌ها و پوشه‌های سایت را نمایش می‌دهد.)

وب سایت: www.jscompress.com کدهای جاوااسکریپت را فشرده می‌کند.

وب سایت: www.jsbutifier.org نیز هستند که برعکس سایت قبلی عمل می‌کند.

(برنامه‌های Advanced IP scanner و Advanced Port Scanner و Nmap برای بدست آوردن پورت و آی‌پی می‌باشد.)

به نرم‌افزارهایی که برای این کار استفاده می‌شود در اصطلاح Packet sniffer گفته می‌شود که Wire shark یکی از آن‌ها به حساب می‌آید.

نرم افزار هویج برای بررسی نفوذ پذیر بودن یک سایت در برابر Sql injection مناسب است

افزونه هک بار.

یادآوری:

[i] روترها، پکت‌ها را به پورت‌ها و پروتکل‌هایی که برنامه‌ی شما نیاز دارد، می‌رساند. برخی آسیب پذیری‌ها مربوط به TCPIP می‌تواننددر روترها بلوکه شود.

[ii] پروتکل‌ها و پورت‌هایی که برنامه شما استفاده نمیکند را بلاک می‌کند. همینطور فایروال‌ها از ترافیک‌هایی که مرتبط با برنامه‌ی شما نباشد از طریق مفهوم فیلترینگ جلوگیری می‌کند.

[iii] سوئیچ‌ها برای جداسازی بخش‌های مختلف یک شبکه استفاده می‌شود. بیش‌ترین محل‌هایی که ترافیک شبکه را به خود اختصاص می‌دهند سوئیچ‌ها هستند

[iv] Credential: هر عاملی که بتوان از آن طریق یک کاربر مجاز را احرازهویت کرد Credential یا گواهینامه گفته می‌شود. مانند نام کاربری و رمز عبور، کارت هوشمند، Certificate و...

[v] MAC address: آدرس مک یا آدرس فیزیکی به کد یکتای هر کارت شبکه که توسط شرکت سازنده و به صورت دیجیتالی بروی آن درج شده گفته می‌شود. با تایپِ دستور ipconfig /all در CMD می‌توانید فیزیکال آدرس مربوط به کارت شبکه ها را مشاهده کنید. این کد چیزی شبیه به کد زیر است:

00-21-91-56-9B-5A

بسیاری از دستگاه‌های شبکه بخشی به نام Mac filtering دارند که می‌توان در آنجا آدرس مک دستگاه‌هایی که مجاز به استفاده از این دستگاه هستند را وارد می‌کند بنابراین هر دستگاهه که آدرس مک آن در این لیست نباشد امکان استفاده از این دستگاه را ندارد. بطور مثال در Access point(AP)ها در کنترلپنل تنظیمات خود چنین امکانی را دارند. نکاتی در مورد کد آدرس مک:

- شش بخشی است
- هر بخش دو قسمت است
- هر بخش بین ۱ تا F یعنی ۱۵(چون در مبنای ۱۶ است) می‌باشد
- هر رقم ۴ بیت است
- مک مجموعا ۴۸ بیتی است (۶*۲*۴)

[vi] استانداردی است که داده‌های ردوبدل شده بین دو کامپیوتر را امن می‌کند، این استاندارد وابسته به پلت فرم خاصی نیست. برای حفظ امنیت از دو پرتکل زیر استفاده می‌کند:

Encapsulating security payload: که اصل داده یا Payload را رمزگذاری می‌کند.

Authentication header: برای احرازهویت کردن کامپیوتر ارسلل کننده و دریافت کننده استفاده می‌شود.

[vii] پروتکلی است که اطلاعات رد و بدل شده بین کلاینت و سرور را رمزگذاری می‌کند.

viii فایلی که حاوی یک سری کد خط فرمانی است که به صورت دسته‌ای (batch) اجرا می‌شود.مثال: با استفاده از حلقه یک فایل bat بسازید که با اجرای آن ۱۰۰ پوشه در درایو C کاربر ایجاد شود و در هر پوشه فایلی ایجاد شود بنام hacked.txt و داخل آن نوشته شده باشد: YOU ARE HACKED. کد زیر را در notepad نئشتهو با پسوند bat ذخیره کنید:

```
for /L %%a IN (1,1,100) do (
md C:\%%a
echo > c:\%%a\hack.txt You Are hacked )
```