5. Разглашение информации (Information Disclosure)

Атаки данного класса направлены на получение дополнительной информации о Web-приложении. Используя эти уязвимости, злоумышленник может определить используемые дистрибутивы ПО, номера версий клиента и сервера и установленные обновления. В других случаях, в утекающей информации может содержаться расположение временных файлов или резервных копий. Во многих случаях эти данные не требуются для работы пользователя. Большинство серверов предоставляют доступ к чрезмерному объему данных, однако необходимо минимизировать объем служебной информации. Чем большими знаниями о приложении будет располагать злоумышленник, тем легче ему будет скомпрометировать систему.

5.1. Индексирование директорий (Directory Indexing)

Предоставление списка файлов в директории представляет собой нормальное поведение Web-сервера, если страница, отображаемая по умолчанию (index.html/home.html/default.htm) отсутствует.

Когда пользователь запрашивает основную страницу сайта, он обычно указываете доменное имя сервера без имени конкретного файла (http://www.example). Сервер просматривает основную папку, находит в ней файл, используемый по умолчанию, и на его основе генерирует ответ. Если такой файл отсутствует, в качестве ответа может вернуться список файлов в директории сервера.

Эта ситуация аналогична выполнению команды "ls" (Unix) или "dir" (Windows) на сервере и форматированию результатов в виде HTML.

В этой ситуации злоумышленник может получить доступ к данным, не предназначенным для свободного доступа. Довольно часто администраторы полагаются на "безопасность через сокрытие", предполагая, что раз гиперссылка на документ отсутствует, то он недоступен непосвященным. Современные сканеры уязвимостей, такие как Nikto, могут динамически добавлять файлы и папки к списку сканируемых в зависимости от результатов запросов. Используя содержимое /robots.txt или полученного списка директорий сканер может найти спрятанное содержимое или другие файлы.

Таким образом, внешне безопасное индексирование директорий может привести к утечке важной информации, которая в дальнейшем будет использована для проведения атак на систему.

Пример:

Используя индексирование директорий можно получить доступ к следующим данным:

- резервные копии (.bak, .old or .orig);
- временные файлы. Такие файлы должны удаляться сервером автоматически, но иногда остаются доступными.
- спрятанные файлы, название которых начинается с символа ".";
- соглашение об именах. Эта информация может помочь предсказать имена файлов или директорий (admin или Admin, back-up или backup).
- список пользователей сервера. Довольно часто для каждого из пользователей создается папка с именем, основанном на названии учетной записи.
- имена файлов конфигурации (.conf, .cfg or .config)
- содержимое серверных сценариев или исполняемых файлов в случае неверно указанных расширений или разрешений.

Могут использоваться три основных сценария получения списка файлов:

1) Ошибки конфигурации. Подобные проблемы возникают, когда администратор ошибочно указывает в конфигурации сервера эту опцию. Подобные ситуации часто возникают при настройке сложных конфигураций, где некоторые папки должны быть доступны для просмотра. С точки зрения злоумышленника запрос не отличается от указанного раньше. Он просто обращается к директории и анализирует результат. Его не беспокоит, почему сервер ведет себя подобным образом.

2) Некоторые компоненты Web-сервера позволяют получать список файлов, даже если это не разрешено в конфигурационных файлах. Обычно это возникает в результате ошибок реализации, когда сервер генерирует список файлов при получении определенного запроса.

3) Базы данных поисковых машин (Google, Wayback machine) могут содержать кэш старых вариантов сервера, включая списки файлов.

Ссылки:

Directory Indexing Vulnerability Alerts
http://www.securityfocus.com/bid/1063
http://www.securityfocus.com/bid/6721
http://www.securityfocus.com/bid/8898

Nessus "Remote File Access" Plugin Web page
http://cgi.nessus.org/plugins/dump.php3?family=Remote%20file%20access

Web Site Indexer Tools
http://www.download-freeware-shareware.com/Internet.php?Theme=112

Intrusion Prevention for Web
http://www.modsecurity.org

Search Engines as a Security Threat
http://it.korea.ac.kr/class/2002/software/Reading%20List/Search%20Engines%20as%20a%20Security%20Threat.pdf

The Google Hacker's Guide
http://johnny.ihackstuff.com/security/premium/The_Google_Hackers_Guide_v1.0.pdf

5.2. Идентификация приложений (Web Server/Application Fingerprinting)

Определение версий приложений используется злоумышленником для получения информации об используемых сервером и клиентом операционных системах, Web-северах и браузерах. Также эта атака может быть направлена на другие компоненты Web-приложения, например, службу каталога или сервер баз данных или используемые технологии программирования.

Обычно подобные атаки осуществляются путем анализа различной информации, предоставляемой Web-сервером, например:

Особенности реализации протокола HTTP;
Заголовки HTTP-ответов;
Используемые сервером расширения файлов (.asp или. jsp);
Значение Cookie (ASPSESSION и т.д.);
Сообщения об ошибках;
Структура каталогов и используемое соглашение об именах (Windows/Unix);
Интерфейсы поддержки разработки Web-приложений(Frontpage/WebPublisher);
Интерфейсы администрирования сервера (iPlanet/Comanche);
Определение версий операционной системы.

Для определения версий клиентских приложений обычно используется анализ HTTP-запросов (порядок следования заголовков, значение User-agent и т.д.). Однако, для этих целей могут

применятся и другие техники. Так, например, анализ заголовков почтовых сообщений, созданных с помощью клиента Microsoft Outlook, позволяет определить версию установленного на компьютере браузера Internet Explorer.

Наличие детальной и точной информации об используемых приложениях очень важно для злоумышленника, поскольку реализация многих атак (например, переполнения буфера) специфично для каждого варианта операционной системы или приложения. Кроме того, детальная информация об инфраструктуре позволяет снизить количество ошибок, и как следствие - общий «шум», производимый атакующим. Данный факт отмечен в HTTP RFC 2068, рекомендующим чтобы значение заголовка Server HTTP ответа являлся настраиваемым параметром.

Примеры:

Сообщения об ошибках – ошибка 404 сервером Apache обозначается фразой "Not Found", в то время как IIS 5.0 отвечает сообщением "Object Not Found".

```
# telnet target1.com 80
Trying target1.com...
Connected to target1.com.
Escape character is '^]'.
HEAD /non-existent-file.txt HTTP/1.0

HTTP/1.1 404 Not Found
Date: Mon, 07 Jun 2004 14:31:03 GMT
Server: Apache/1.3.29 (Unix)
mod_perl/1.29
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Синтаксис заголовков также может отличаться. Например, использование строчных или заглавных букв в названии параметров ("Content-Length" в IIS или "Content-length" в Netscape-Enterprise/6.0).

Не смотря на требования HTTP RFC, существуют семантические особенности при генерации заголовков различными серверами. Например, Apache передает параметр Date перед значением заголовка Server, в то время как IIS использует обратный порядок. Порядок значений параметров так же может отличаться. Например, при обработке запроса OPTIONS Apache возвращает только параметр Allow, в то время как IIS дополнительно включает параметр Public.

Аналогичным образом может анализироваться наличие опциональных заголовков (Vary, Expires и т.д.) и реакция сервера на неверные запросы ("GET //", "GET/%2f" и т.д.).

Ссылки

"An Introduction to HTTP fingerprinting"
http://net-square.com/httprint/httprint_paper.html

"Hypertext Transfer Protocol -- HTTP/1.1"
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2068.html#sec-14.39

"HMAP: A Technique and Tool for Remote Identification of HTTP Servers"
http://seclab.cs.ucdavis.edu/papers/hmap-thesis.pdf

"Identifying Web Servers: A first-look into Web Server Fingerprinting"
http://www.blackhat.com/presentations/bh-asia-02/bh-asia-02-grossman.pdf

"Mask Your Web Server for Enhanced Security"
http://www.port80software.com/support/articles/maskyourwebserver

"Web Intrusion Detection and Prevention"
http://www.modsecurity.org

"IIS LockDown Tool 2.1"
http://www.microsoft.com/downloads/details.aspx?FamilyID=DDE9EFC0-BB30-47EB-9A61-FD755D23CDEC&displaylang=en

"URLScan Tool"
http://www.microsoft.com/downloads/details.aspx?FamilyID=f4c5a724-cafa-4e88-8c37-c9d5abed1863&DisplayLang=en

"ServerMask Tool"
http://www.port80software.com/products/servermask/

5.3. Утечка информации (Information Leakage)

Эти уязвимости возникают в ситуациях, когда сервер публикует важную информацию, например комментарии разработчиков или сообщения об ошибках, которая может быть использована для компрометации системы. Ценные с точки зрения злоумышленника данные могут содержаться в комментариях HTML, сообщениях об ошибках или просто присутствовать в открытом виде. Существует огромное количество ситуаций, в которых может произойти утечка информации. Не обязательно она приводит к возникновению уязвимости, но часто дает атакующему прекрасное пособие для развития атаки. С утечкой важной информации могут возникать риски различной степени, поэтому необходимо минимизировать количество служебной информации, доступной на клиентской стороне.

Анализ доступной информации позволяет злоумышленнику произвести разведку и получить представление о структуре директорий сервера, используемых SQL запросах, названиях ключевых процессов и программ сервера.

Часто разработчики оставляют комментарии в HTML страницах и коде сценариев для облегчения поиска ошибок и поддержки приложения. Эта информация может варьироваться от простых описаний деталей функционирования программы до, в худших случаях, имен пользователей и паролей, используемых при отладке.

Утечка информации может относиться и к конфиденциальным данным, обрабатываемым сервером. Это могут быть идентификаторы пользователя (ИНН, номера водительских удостоверений, паспортов и т.д.), а также текущая информация (баланс лицевого счета или история платежей).

Многие атаки этой категории выходят за рамки защиты Web-приложений и переходят в область физической безопасности. Утечка информации в этом случае часто возникает, когда в браузере отображается информация, которая не должна выводиться в открытом виде даже пользователю. В качестве примера можно привести пароли пользователя, номера кредитных карточек и т.д.

Примеры:

Далее рассматриваются три основных категории утечки информации: комментарии разработчиков, сообщения об ошибках и отображение конфиденциальной информации.

Комментарии в коде.

```
<TABLE border="0" cellPadding="0" cellSpacing="0" height="59" width="591">
      <TBODY>
        <TR>
            <!--If the image files are missing, restart VADER -->
            <TD bgColor="#ffffff" colSpan="5" height="17" width="587"> </TD>
         </TR>
```

Выше приведен комментарий разработчиков или тестировщиков, который указывает на то, что в случае проблем с загрузкой изображений необходимо перезагрузить сервер VADER.

Подробные сообщения об ошибках могут возникать в результате специально сформированного запроса. Ниже приведен пример сообщения, типичного для ошибки, возникающей в результате SQL-запроса. Для реализации атаки с внедрением кода SQL обычно требуется знание структуры запросов, осуществляемых Web-сервером. Информация, передаваемая в подробной информации об ошибке может быть использована для построения атакующим корректных запасов к серверу баз данных.

Ниже приведено сообщение, выдаваемое сервером при вводе символа апострофа в качестве имени пользователя.

```
An Error Has Occurred.
Error Message:
System.Data.OleDb.OleDbException:
Syntax error
(missing operator) in query expression
'username = '''
and password = 'g'. at
System.Data.OleDb.OleDbCommand.
ExecuteCommandTextErrorHandling (
Int32 hr) at
System.Data.OleDb.OleDbCommand.
ExecuteCommandTextForSingleResult (
tagDBPARAMS dbParams,  Object&
executeResult) at
```

В первой части сообщения выводится часть запроса, вызвавшего ошибку. По этой информации злоумышленник может получить информацию об используемых параметрах запроса и месте запроса, в котором осуществляется внедрение кода.

Ссылки:

"Best practices with custom error pages in .Net", Microsoft Support
http://support.microsoft.com/default.aspx?scid=kb;en-us;834452

"Creating Custom ASP Error Pages", Microsoft Support
http://support.microsoft.com/default.aspx?scid=kb;en-us;224070

"Apache Custom Error Pages", Code Style
http://www.codestyle.org/sitemanager/apache/errors-Custom.shtml

"Customizing the Look of Error Messages in JSP", DrewFalkman.com
http://www.drewfalkman.com/resources/CustomErrorPages.cfm

ColdFusion Custom Error Pages
http://livedocs.macromedia.com/coldfusion/6/Developing_ColdFusion_MX_Applications_with_CFML/Errors6.htm

Obfuscators: JAVA
http://www.cs.auckland.ac.nz/~cthombor/Students/hlai/hongying.pdf

5.4. Обратный путь в директориях (Path Traversal)

Данная техника атак направлена на получения доступа к файлам, директориям и командам, находящимся вне основной директории Web-сервера. Злоумышленник может манипулировать параметрами URL с целью получить доступ к файлам или выполнить команды, располагаемые в файловой системе Web-сервера. Для подобных атак потенциально уязвимо любое устройство, имеющее Web-интерфейс.

Многие Web-серверы ограничивают доступ пользователя определенной частью файловой системы, обычно называемой "web document root" или "CGI root". Эти директории содержат

файлы, предназначенные для пользователя и программы, необходимые для получения доступа к функциям Web-приложения.

Большинство базовых атак, эксплуатирующих обратный путь, основаны на внедрении в URL символов "../", для того, чтобы изменить расположение ресурса, который будет обрабатываться сервером. Поскольку большинство Web-серверов фильтруют эту последовательность, злоумышленник может воспользоваться альтернативными кодировками для представления символов перехода по директориям. Популярные приемы включают использование альтернативных кодировок, например Unicode ("..%u2216" или "..%c0%af"), использование обратного слеша ("..\") в Windows-серверах, символов URLEncode ("%2e%2e%2f") или двойная кодировка URLEncode ("..%255c").

Даже если Web-сервер ограничивает доступ к файлам определенным каталогом, эта уязвимость может возникать в сценариях или CGI-программах. Возможность использования обратного пути в каталогах довольно часто возникает в приложениях, использующих механизмы шаблонов или загружают их текст страниц из файлов на сервере. В этом варианте атаки злоумышленник модифицирует имя файла, передаваемое в качестве параметра CGI-программы или серверного сценария. В результате злоумышленник может получить исходный код сценариев. Довольно часто к имени запрашиваемого файла добавляются специальные символы, такие как "%00", с целью обхода фильтров.

Примеры:

Обратный путь в каталогах Web-сервера

http://example/../../../../../some/file
http://example/..%255c..%255c..%255csome/file
http://example/..%u2216..%u2216some/file

Обратный путь в каталогах Web-приложения

Исходный URL: http://example/foo.cgi?home=index.htm
Атака: http://example/foo.cgi?home=foo.cgi

В приведенном сценарии Web-приложение генерирует страницу, содержащую исходный код сценария foo.cgi, поскольку значение переменной home используется как имя загружаемого файла. Обратите внимание, что в данном случае злоумышленник не использует специальных символов, поскольку целью является файл в той же директории, в которой располагается файл index.htm.

Обратный путь в каталогах Web-приложения с использованием специальных символов:

Исходный URL: http://example/scripts/foo.cgi?page=menu.txt
Атака: http://example/scripts/foo.cgi?page=../scripts/foo.cgi%00txt

В приведенном примере Web-приложение загружает исходный текст сценария foo.cgi. Атакующий использует символы "../" для перехода на уровень выше по дереву каталогов и перехода в директорию /scripts. Символ "%00" используется для обхода проверки расширения файла (приложение позволяет обращаться только к файлам .txt)и для того, чтобы расширение не использовалось при загрузке файла.

Ссылки:

"CERT© Advisory CA-2001-12 Superfluous Decoding Vulnerability in IIS"
http://www.cert.org/advisories/CA-2001-12.html

"Novell Groupwise Arbitrary File Retrieval Vulnerability"
http://www.securityfocus.com/bid/3436/info/

5.5. Предсказуемое расположение ресурсов (Predictable Resource Location)

Предсказуемое расположение ресурсов позволяет злоумышленнику получить доступ к скрытым данным или функциональным возможностям. Путем подбора злоумышленник может получить доступ к содержимому, не предназначенному для публичного просмотра. Временные файлы, файлы резервных копий, файлы конфигурации или стандартные примеры часто являются целью подобных атак. В большинстве случаев перебор может быть оптимизирован путем использования стандартного соглашения об именах файлов и директорий сервера. Получаемые злоумышленником файлы могут содержать информацию о дизайне приложения, информацию из баз данных, имена машин или пароли, пути к директориям. Также «скрытые» файлы могут содержать уязвимости, отсутствующие в основном приложении. На эту атаку часто ссылаются как на перечисление файлов и директорий (Forced Browsing, File Enumeration, Directory Enumeration).

Пример:

Атакующий может создать запрос к любому файлу или папке на сервере. Наличие или отсутствие ресурса определяется по коду ошибки (например, 404 в случае отсутствия папки или 403 в случае её наличия на сервере). Ниже приведены варианты подобных запросов.

Слепой поиск популярных названий директорий:

/admin/
/backup/
/logs/
/vulnerable_file.cgi

Изменение расширений существующего файла: (/test.asp)

/test.asp.bak
/test.bak
/test