

6. Логические атаки (Logical Attacks)

Атаки данного класса направлены на эксплуатацию функций приложения или логики его функционирования. Логика приложения представляет собой ожидаемый процесс функционирования программы при выполнении определенных действий. В качестве примеров можно привести восстановление пролей, регистрацию учетных записей, аукционные торги, транзакции в системах электронной коммерции. Приложение может требовать от пользователя корректного выполнения нескольких последовательных действий для выполнения определенной задачи. Злоумышленник может обойти или использовать эти механизмы в своих целях.

6.1. Злоупотребление функциональными возможностями (Abuse of Functionality).

Данные атаки направлены на использование функций Web-приложения с целью обхода механизмов разграничения доступа. Некоторые механизмы Web-приложения, включая функции обеспечения безопасности, могут быть использованы для этих целей. Наличие уязвимости в одном из, возможно, второстепенных компонентов приложения может привести к компрометации всего приложения. Уровень риска и потенциальные возможности злоумышленника в случае проведения атаки очень сильно зависят от конкретного приложения.

Злоупотребление функциональными возможностями очень часто используется совместно с другими атаками, такими как обратный путь в директориях и т.д. К примеру, при наличии уязвимости типа межсайтовое выполнение сценариев в HTML-чате злоумышленник может использовать функции чата для рассылки URL, эксплуатирующий уязвимость, всем текущим пользователям.

С глобальной точки зрения, все атаки на компьютерные системы являются злоупотреблениями функциональными возможностями. Особенно это относится к атакам, направленным на Web-приложения, которые не требуют модификации функций программы.

Пример:

Примеры злоупотребления функциональными возможностями включают в себя:

- Использование функций поиска для получения доступа к файлам за пределами корневой директории Web-сервера;
- Использование функции загрузки файлов на сервер для перезаписи файлов конфигурации или внедрения серверных сценариев;
- Реализация отказа в обслуживании путем использования функции блокировки учетной записи при многократном вводе неправильного пароля.

Ниже приведены реальные примеры подобных уязвимостей, взятые из реальной жизни.

Программа Matt Wright FormMail

Программа "FormMail" представляет собой приложение на языке PERL, используемое для передачи данных из HTML-формы на указанный почтовый адрес. Этот сценарий довольно удобно использовать для организации функции обратной связи на сервере. К сожалению, эта программа предоставляла злоумышленнику возможность передавать почтовые сообщения любому почтовому пользователю. Таким образом, приложение могло быть использовано в качестве почтового ретранслятора для рассылки спама.

Злоумышленник использовал параметры URL GET-запроса для указания получателя почтового сообщения, к примеру:

```
http://example/cgi-bin/FormMail.pl?  
recipient=email@victim.example&message=you%20got%20spam
```

В качестве отправителя почтового сообщения указывался адрес Web-сервера, что позволяло злоумышленнику оставаться полностью анонимным.

Macromedia's Cold Fusion

Иногда базовый интерфейс администрирования, поставляемый вместе с Web-приложением, может использоваться с непредусмотренными разработчиками целями. К примеру, Macromedia's Cold Fusion по умолчанию имеет модуль, позволяющий просматривать исходный код сценариев. Злоупотребление этой функцией может привести к получению критичной информации Web-приложения. Удаление или отключение этой функции весьма проблематично, поскольку от него зависят важные компоненты приложения.

Модификация цены в Smartwin CyberOffice

Иногда изменение данных, обрабатываемых приложением, может позволить модифицировать поведение программы. К примеру, уязвимость в функции покупки приложения CyberOffice позволяла модифицировать значение цены, передаваемой пользователю в скрытом поле HTML-формы. Страница подтверждения заказа загружалась злоумышленником, модифицировалась на клиенте и передавалась серверу уже с модифицированным значением цены.

Ссылки:

"FormMail Real Name/Email Address CGI Variable Spamming Vulnerability"

<http://www.securityfocus.com/bid/3955>

"CVE-1999-0800"

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0800>

"CA Unicenter pdmcgi.exe View Arbitrary File"

http://www.osvdb.org/displayvuln.php?osvdb_id=3247

"PeopleSoft PeopleBooks Search CGI Flaw"

http://www.osvdb.org/displayvuln.php?osvdb_id=2815

"iisCART2000 Upload Vulnerability"

<http://secunia.com/advisories/8927/>

"PROTEGO Security Advisory #PSA200401"

<http://www.protego.dk/advisories/200401.html>

"Price modification possible in CyberOffice Shopping Cart"

<http://archives.neohapsis.com/archives/bugtraq/2000-10/0011.html>

6.2. Отказ в обслуживании (Denial of Service).

Данный класс атак направлен на нарушение доступности Web-сервера. Обычно атаки, направленные на отказ в обслуживании реализуются на сетевом уровне, однако они могут быть направлены и на прикладной уровень. Используя функции Web-приложения, злоумышленник может исчерпать критичные ресурсы системы, или воспользоваться уязвимостью, приводящий к прекращению функционирования системы.

Обычно DoS атаки направлены на исчерпание критичных системных ресурсов, таких как вычислительные мощности, оперативная память, дисковое пространство или пропускная способность каналов связи. Если какой-то из ресурсов достигнет максимальной загрузки, приложение целиком будет недоступно.

Атаки могут быть направлены на любой из компонентов Web-приложения, например, такие как сервер СУБД, сервер аутентификации и т.д. В отличие от атак на сетевом уровне, требующих значительных ресурсов злоумышленника, атаки на прикладном уровне обычно легче реализовать.

Примеры:

Предположим, что сервер Health-Care генерирует отчеты о клинической истории пользователей. Для генерации каждого отчета сервер запрашивает все записи, соответствующие определенному номеру социального страхования. Поскольку в базе содержатся сотни миллионов записей, пользователю приходится ожидать результата несколько минут. В это время загрузка процессора сервера СУБД достигает 60%.

Злоумышленник может послать десять одновременных запросов на получение отчетов, что с высокой вероятностью приведет к отказу в обслуживании, поскольку загрузка процессора сервера баз данных достигнет максимального значения. На время обработки запросов злоумышленника нормальная работа сервера будет невозможна.

DoS на другой сервер

Злоумышленник может разместить на популярном Web-форуме ссылку (например, в виде изображения в сообщении) на другой ресурс. При заходе на форум, пользователи будут автоматически загружать данные с атакуемого сервера указанный ресурс, используя его ресурсы. Если на атакуемом сервере используется система предотвращения атак с функцией блокировки IP-адреса атакующего, в ссылке может использоваться сигнатура атаки (например ../../../../etc/passwd), что приведет к блокировке пользователей, зашедших на форум.

Атаки на сервер СУБД

Злоумышленник может воспользоваться внедрением кода SQL для удаления данных из таблиц, что приведет к отказу в обслуживании приложения.

6.3. Недостаточное противодействие автоматизации (Insufficient Anti-automation)

Недостаточное противодействие автоматизации возникает, когда сервер позволяет автоматически выполнять операции, которые должны проводиться вручную. Для некоторых функций приложения необходимо реализовывать защиту от автоматических атак.

Автоматизированные программы могут варьироваться от безобидных роботов поисковых систем до систем автоматизированного поиска уязвимостей и регистрации учетных записей. Подобные роботы генерируют тысячи запросов в минуту, что может привести к падению производительности всего приложения.

Противодействие автоматизации заключается в ограничении возможностей подобных утилит. Например, файл robots может предотвращать индексирование некоторых частей сервера, а дополнительные средства идентификации предотвращать автоматическую регистрацию сотен учетных записей системы электронной почты.

Ссылки

Telling Humans Apart (Automatically)

<http://www.captcha.net/>

"Ravaged by Robots!", By Randal L. Schwartz

<http://www.webtechniques.com/archives/2001/12/perl/>

".Net Components Make Visual Verification Easier", By JingDong (Jordan) Zhang

<http://go.cadwire.net/?3870,3,1>

"Vorras Antibot"

<http://www.vorras.com/products/antibot/>

"Inaccessibility of Visually-Oriented Anti-Robot Tests"

<http://www.w3.org/TR/2003/WD-turingtest-20031105/>

6.4. Недостаточная проверка процесса (Insufficient Process Validation)

Уязвимости этого класса возникают, когда сервер не достаточно проверяет последовательность выполнения операций приложения. Если состояние сессии пользователя и приложения должным образом не контролируется, приложение может быть уязвимо для мошеннических действий.

В процессе доступа к некоторым функциям приложения ожидается, что пользователь выполнит ряд действий в определенном порядке. Если некоторые действия выполняются неверно или в неправильном порядке, возникает ошибка, приводящая к нарушению целостности. Примерами подобных функций выступают переводы, восстановление паролей, подтверждение покупки, создание учетной записи и т.д. В большинстве случаев эти процессы состоят из ряда последовательных действий, осуществляемых в четком порядке.

Для обеспечения корректной работы подобных функций Web-приложение должно четко отслеживать состояние сессии пользователя и отслеживать её соответствие текущим операциям. В большинстве случаев это осуществляется путем сохранения состояния сессии в cookie или скрытом поле формы HTML. Но поскольку эти значения могут быть модифицированы пользователем, обязательно должна проводиться проверка этих значений на сервере. Если этого не происходит, злоумышленник получает возможность обойти последовательность действий, и как следствие - логику приложения.

Пример:

Система электронной торговли может предлагать скидку на продукт В, в случае покупки продукта А. Пользователь, не желающий покупать продукт А, может попытаться приобрести продукт В со скидкой. Заполнив заказ на покупку обоих продуктов, пользователь получит скидку. Затем пользователь возвращается к форме подтверждения заказа и удаляет продукт А из покупаемых, путем модификации значений в форме. Если сервер повторно не проверит возможность покупки продукта В по указанной цене без продукта А, будет осуществлена закупка по низкой цене.

Ссылки:

"Dos and Don'ts of Client Authentication on the Web", Kevin Fu, Emil Sit, Kendra Smith, Nick Feamster - MIT Laboratory for Computer Science

<http://cookies.lcs.mit.edu/pubs/webauth:tr.pdf>