2. Авторизация (Authorization)

Данный раздел посвящен атакам, направленным на методы, которые используются Web-сервером для определения того, имеет ли пользователь, служба или приложение необходимые для совершения действия разрешения. Многие Web-сайты разрешают только определенным пользователям получать доступ к некоторому содержимому или функциям приложения. Доступ другим пользователям должен быть ограничен. Используя различные техники, злоумышленник может повысить свои привилегии и получить доступ к защищенным ресурсам.

2.1. Предсказуемое значение идентификатора сессии (Credential/Session Prediction)

Предсказуемое значение идентификатора сессии позволяет перехватывать сессии других пользователей. Подобные атаки выполняются путем предсказания или угадывания уникального идентификатора сессии пользователя. Эта атака также как и перехват сессии (Session Hijacking) в случае успеха позволяет злоумышленнику послать запрос Web-серверу с правами скомпрометированного пользователя. Дизайн многих серверов предполагает аутентификацию пользователя при первом обращении и дальнейшее отслеживание его сессии. Для этого пользователь указывает комбинацию имени и пароля. Вместо повторной передачи имени пользователя и пароля при каждой транзакции, Web-сервер генерирует уникальный идентификатор, который присваивается сессии пользователя. Последующие запросы пользователя к серверу содержат идентификатор сессии как доказательство того, что аутентификация была успешно пройдена. Если атакующий может предсказать или угадать значение идентификатора другого пользователя, это может быть использовано для проведения атаки.

Примеры:

Многие серверы генерируют идентификаторы сессии, используя алгоритмы собственной разработки. Подобные алгоритмы могут просто увеличивать значение идентификатора для каждого запроса пользователя. Другой распространенный вариант – использование функции от текущего времени или других специфичных для компьютера данных.

Идентификатор сессии сохраняется в cookie, скрытых полях форм или URL. Если атакующий имеет возможность определить алгоритм, используемый для генерации идентификатора сессии, он может выполнить следующие действия:

1) подключиться к серверу, используя текущий идентификатор сессии;
2) вычислить или подобрать следующий идентификатор сессии;
3) присвоить полученное значение идентификатора cookie/скрытому полю формы/URL.

Ссылки:

"iDefense: Brute-Force Exploitation of Web Application Session ID's", By David Endler - iDEFENSE Labs
http://www.cgisecurity.com/lib/SessionIDs.pdf

"Best Practices in Managing HTTP-Based Client Sessions", Gunter Ollmann - X-Force Security Assessment Services EMEA
http://www.technicalinfo.net/papers/WebBasedSessionManagement.html

"A Guide to Web Authentication Alternatives", Jan Wolter
http://www.unixpapa.com/auth/homebuilt.html

2.2. Недостаточная авторизация (Insufficient Authorization)

Недостаточная авторизация возникает, когда Web-сервер позволяет атакующему получать доступ к важной информации или функциям, доступ к которым должен быть ограничен. То, что пользователь прошел аутентификацию не означает, что он должен получить доступ ко всем функциям и содержимому сервера. Кроме аутентификации должно быть реализовано разграничение доступа.

Процедура авторизации определяет, какие действия может совершать пользователь, служба или приложение. Правильно построенные правила доступа должны ограничивать действия пользователя согласно политике безопасности. Доступ к важным ресурсам сайта должен быть разрешен только администраторам.

Примеры:

В прошлом многие Web-серверы сохраняли важные ресурсы в "скрытых" директориях, таких как "/admin" или "/log". Если атакующий запрашивал эти ресурсы напрямую, он получал к ним доступ и мог перенастроить сервер, получить доступ к важной информации либо полностью скомпрометировать систему.

Некоторые серверы, после аутентификации, сохраняют в cookie или скрытых полях идентификатор "роли" пользователя в рамках Web-приложения. Если разграничение доступа основывается на проверке данного параметра без верификации принадлежности к роли при каждом запросе, злоумышленник может повысить свои привилегии, просто модифицировав значение cookie.

К примеру, значение cookie
SessionId=12345678;Role=User

Заменяется на
SessionId=12345678;Role=Admin

Ссылки:

"Brute Force Attack", Imperva Glossary

http://www.imperva.com/application_defense_center/glossary/brute_force.html

"iDefense: Brute-Force Exploitation of Web Application Session ID's", By David Endler - iDEFENSE Labs
http://www.cgisecurity.com/lib/SessionIDs.pdf

2.3. Отсутствие таймаута сессии (Insufficient Session Expiration)

В случае если для идентификатора сессии или учетных данных не предусмотрен таймаут или его значение слишком велико, злоумышленник может воспользоваться старыми данными для авторизации. Это повышает уязвимость сервера для атак, связанных с кражей идентификационных данных. Поскольку протокол HTTP не предусматривает контроль сессии, Web-серверы обычно используют идентификаторы сессии для определения запросов пользователя. Таким образом, конфиденциальность каждого идентификатора должна быть обеспечена, чтобы предотвратить множественный доступ пользователей с одной учетной записью. Похищенный идентификатор может использоваться для доступа к данным пользователя или осуществления мошеннических транзакций. Отсутствие таймаута сессии увеличивает вероятность успеха различных атак. К примеру, злоумышленник может получить идентификатор сессии, используя сетевой анализатор или уязвимость типа межсайтовое выполнение сценариев. Хотя таймаут не поможет в случае, если идентификатор будет использован немедленно, ограничение времени действия поможет в случае более поздних попыток использования идентификатора.

В другой ситуации, если пользователь получает доступ к серверу с публичного компьютера (библиотека, Internet-кафе и т.д.) отсутствие таймаута сессии может позволить злоумышленнику воспользоваться историей браузера для просмотра страниц пользователя.

Большое значение таймаута увеличивает шансы подбора действующего идентификатора. Кроме того, увеличение этого параметра ведет к увеличению одновременно открытых сессий, что ещё больше повышает вероятность успешного подбора.

Пример:

При использовании публичного компьютера, когда несколько пользователей имеют неограниченный физический доступ к машине, отсутствие таймаута сессии позволяет злоумышленнику просматривать страницы, посещенные другим пользователем. Если функция выхода из системы просто перенаправляет на основную страницу Web-сервера, а не завершает сессию, страницы, посещенные пользователем, могут быть просмотрены злоумышленником. Поскольку идентификатор сессии не был отмечен как недействительный, атакующий получит доступ к страницам сервера без повторной аутентификации.

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

2.4. Фиксация сессии (Session Fixation)

Используя данный класс атак, злоумышленник присваивает идентификатору сессии пользователя заданное значение. В зависимости от функциональных возможностей сервера, существует несколько способов "зафиксировать" значение идентификатора сессии. Для этого могут использоваться атаки типа межсайтовое выполнение сценариев или подготовка сайта с помощью предварительного HTTP запроса. После фиксации значения идентификатора сессии атакующий ожидает момента, когда пользователь войдет в систему. После входа пользователя, злоумышленник использует идентификатор сессии для получения доступа к системе от имени пользователя.

Можно выделить два типа систем управления сессиями на основе идентификаторов. Первый из них, "разрешающий", позволяет браузеру указывать любой идентификатор. Системы второго "строгого" типа обрабатывают только идентификаторы, сгенерированные сервером. Если используются "разрешающие" системы, злоумышленник может выбрать любой идентификатор сессии. В случае со "строгими" серверами злоумышленнику приходится поддерживать "сессию-заглушку" и периодически соединяться с сервером для избежание закрытия сессии по таймауту.

Без наличия активной защиты от фиксации сессии, эта атака может быть использована против любого сервера, аутентифицирующего пользователей с помощью идентификатора сессии. Большинство Web-серверов сохраняет ID в cookie, но это значение так же может присутствовать в URL или скрытом поле формы.

К сожалению, системы, использующие cookie, являются наиболее уязвимыми. Большинство известных на настоящий момент вариантов фиксации сессии направлены именно на значение cookie.

В отличие от кражи идентификатора, фиксация сессии предоставляет злоумышленнику гораздо больший простор для творчества. Это связанно с тем, что активная фаза атаки происходит до входа пользователя в систему.

Пример:

Атаки, направленные на фиксацию сессии обычно проходят в три этапа.

1) Установление сессии

Злоумышленник устанавливает сессию-заглушку на атакуемом сервере и получает от сервера идентификатор или выбирает произвольный идентификатор. В некоторых случаях сессия-заглушка должна поддерживаться в активном состоянии путем периодических обращений к серверу.

2) Фиксация сессии
Злоумышленник передает значение идентификатора сессии-заглушки браузеру пользователя и фиксирует его идентификатор сессии. Это можно сделать, например, установив значение cookie в браузере с помощью XSS.

3) Подключение к сессии
Атакующий ожидает аутентификации пользователя на сервере. После того, как пользователь зашел на сайт, злоумышленник подключается к серверу, используя зафиксированный идентификатор, и получает доступ к сессии пользователя.

Для фиксации ID сессии могут быть использованы различные техники, такие как:

- Установка значения cookie с помощью языков сценариев на стороне клиента.
Если уязвимость типа межсайтовое выполнение сценариев присутствует на любом сервере в домене, злоумышленник получает возможность установить значение cookie на стороне клиента.

Пример кода:
http://example/<script>document.cookie="sessionid=1234;%20domain=.example.dom";</script>.idc

- Установка значения cookie с помощью тега META
Это техника похожа на предыдущую, но может быть использована, когда предприняты меры против внедрения тегов сценариев.

Пример кода:

http://example/<meta%20http-equiv=Set-Cookie%20content="sessionid=1234;%20domain=.example.dom">.idc

Установка cookie с использованием заголовка ответа HTTP
Злоумышленник использует атакуемый сервер или любой сервер в домене для того, чтобы установить cookie с идентификатором сессии.

Это может быть реализовано различными методами, например:
- Взлом сервера в домене (например, слабо администрируемый сервер WAP).
- Подмена значений в кэше DNS-сервера пользователя с целью добавления сервера атакующего в домен.
- Установка ложного WEB-сервера в домене (к примеру, на рабочей станции в среде Active Directory, где все машины в DNS принадлежат одному домену).
- Использование атаки типа расщепление HTTP ответа (response splitting).

Замечание:

Фиксация сессии на продолжительный промежуток времени может быть осуществлена с использованием постоянных cookie (например, со сроком действия 10 лет), которые сохраняются даже после перезагрузки компьютера.

Пример кода: http://example/<script>document.cookie="sessionid=1234;%20Expires=Friday,%201-Jan2010%2000:00:00%20GMT";</script>.idc

Ссылки:

"Session Fixation Vulnerability in Web-based Applications", By Mitja Kolsek - Acros Security
http://www.acrossecurity.com/papers/session_fixation.pdf

"Divide and Conquer", By Amit Klein – Sanctum
http://www.sanctuminc.com/pdf/whitepaper_httpresponse.pdf