



ОТЧЕТ ОТ ТЕСТ ЗА ПРОНИКВАНЕ · СЪСТАВЕН ПРИМЕР

Съставен примерен Тест за проникване

Изготвен за [Клиент] от PXLSecurity LTD.

Това е примерен документ, не предаван на клиент. Той е съставен от няколко анонимизирани ангажимента на PXLSec от различни видове — уеб, API, мобилни, вътрешна мрежа и външен периметър. Всяка констатация е преписана със синтетични, илюстративни данни върху резервирани примерни имена и адреси. Той не описва нито един действителен клиент и в него не присъства действителен клиент, хост, човек или данни за достъп.

НОМЕР НА ОТЧЕТА

PXL-2026-0142

ВЕРСИЯ

1.0 (окончателна)

ДАТА НА ОТЧЕТА

[Дати на ангажимента]

ПЕРИОД НА ТЕСТВАНЕ

[Договорени дати]

КЛАСИФИКАЦИЯ

Конфиденциално — примерен документ

ИЗГОТВЕН ОТ

PXLSecurity LTD., София, България

PXLSecurity LTD. · бул. „Васил Левски“ 12, 1142 София, България · VAT BG203239331 · hello@pxlsec.com · независими от 2014 г.

НАЧАЛНИ СТРАНИЦИ

Контрол на документа

Тази страница отразява какво представлява документът, който го държи и как се е променял. В този примерен документ всяко поле, идентифициращо клиента, е оставено като очевиден заместител.

Заглавие на документа	Съставен примерен тест за проникване — отчет
Номер на отчета	PXL-2026-0142 · версия 1.0 (окончателна) · заменя 0.9 (проект)
Клиент	[Клиент]
Дати на ангажимента	[Договорени дати] · отчет с дата [Дати на ангажимента]
Автор	Старши консултант по сигурността, PXLSecurity LTD.
Осигуряване на качеството	Втори старши консултант — технически преглед и независимо възпроизвеждане на всяка констатация преди издаването
Класификация	Конфиденциално. Този примерен документ не съдържа клиентски данни.

История на редакциите

Версия	Дата	Автор	Промяна
0.1	[Договорени дати]	Водещ консултант	Първоначален проект, издаден в края на активното тестване
0.9	[Договорени дати]	Проверяващ качеството	Технически преглед; всяка констатация възпроизведена независимо
1.0	[Дати на ангажимента]	Водещ консултант	Окончателно издание до списъка за разпространение по-долу

Разпространение

Име	Роля	Организация	Копие
[Лице за контакт на клиента]	Поръчител на ангажимента	[Клиент]	Пълен отчет
[Лице за контакт на клиента]	Ръководител на разработката	[Клиент]	Пълен отчет
Водещ консултант	Автор	PXLSecurity LTD.	Пълен отчет
[Одитор]	Външен одитор	[Одиторска фирма]	Само по указание на клиента

Конфиденциалност. Действителен отчет от този вид описва, с подробност, която позволява възпроизвеждане, как да бъде компрометирана продукционна среда. С него трябва да се работи като с един от най-чувствителните документи, които организацията притежава: шифрован при съхранение и при пренос, споделян само с посочения списък за разпространение и никога публикуван в система за задачи с общо предназначение или в споделено файлово хранилище. Този конкретен документ е примерен и не съдържа клиентски данни, така че нищо от това не се отнася за него — той е публикуван умишлено.

НАЧАЛНИ СТРАНИЦИ

Съдържание

Структурата по-долу е структурата на всеки отчет от тест за проникване на PXLSec. Ръководството може да прочете раздели 1 до 4 и да действа; инженерите могат да работят от раздел 5 нататък, без да четат нищо друго.

РАЗДЕЛ	СТР.
1 Резюме за ръководството	4
Какво беше тествано, основната картина на риска и трите неща, които да поправите първо — написано за нетехнически читател.	
2 Обхват на ангажимента	6
Цели, вид на оценката, период на тестване, стандарти, изключения и допусканията, от които зависи резултатът.	
3 Методология	8
Петте етапа, какво произвежда всеки от тях, как се използват инструментите и правилата за безопасност, при които тестваме.	
4 Оценки на критичността	10
Скалата, какво означава всяко ниво на практика и как CVSS информира оценката, без да я замества.	
5 Обобщение на констатациите	11
Всяка констатация с нейния идентификатор, критичност и текущ статус, в реда на отстраняване.	
6 Подробни констатации	12
По един раздел за всяка констатация: описание, доказателство, възпроизвеждане, въздействие и конкретна препоръка.	
7 Повторен тест и приключване	24
Какво покрива включеният повторен тест и какво заявява писмото за приключване.	
A Приложение А — Използвани инструменти	25
Инструментите в ангажимента и разделението на труда между автоматизацията и ръчната работа.	
B Приложение Б — Извън обхвата	26
Какво беше изрично изключено и какво означава това за увереността, която този отчет дава.	

C

Приложение В — Как да четете този отчет

27

Кой кой раздел да прочете и как да го превърне в план за отстраняване.

A note on what this sample is. Findings F-01 to F-07 are drawn from several different anonymised engagements and are presented here as if they came from one. No real assessment of a single application would normally surface a mobile bundle, a database cluster, an Active Directory service account and an internet-facing gateway in the same scope. They are collected in one document so that a prospective client can see the full range of work in a single read. Every transcript, hostname, address, account and score is synthetic and illustrative.

РАЗДЕЛ 1

Резюме за ръководството

PXLSecurity LTD. беше нает от [Клиент] да извърши ръчен тест за проникване, воден от цели. Този раздел е написан за нетехнически читател. Той казва какво беше тествано, колко сериозен е резултатът и какво да се направи първо. Нищо в него не изисква познания по сигурност, за да се действа по него.

Какво беше тествано

Старши инженери тестваха системите в обхвата на ръка през договорения период, работейки както би работил истински нападател: намери път навътре, виж докъде стига и го докажи. Съставният обхват зад този примерен документ покриваше обърнато към клиенти уеб приложение и неговото REST API, публикувано мобилно приложение, слоя с базата данни зад приложението, вътрешна мрежа с нейния домейн в Active Directory и външния периметър. Тестването беше с удостоверяване там, където приложението предлагаше акаунти, така че резултатът отразява какво би могъл да достигне влязъл клиент или компрометиран служител, а не само какво вижда анонимен посетител.

Заглавието

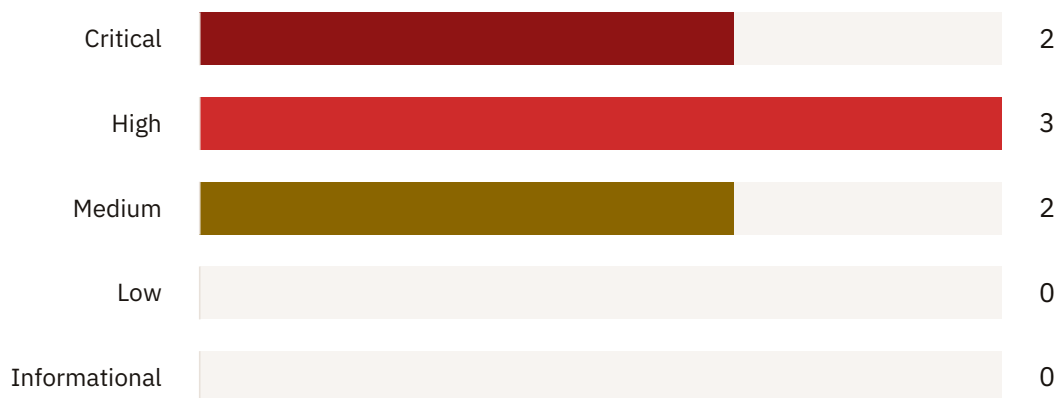
Оценката установи седем констатации: две Critical, три High и две Medium. Общата оценка на риска за ангажимента е High.

7 констатации	2 Critical	3 High	2 Medium	High обща оценка на риска
------------------	---------------	-----------	-------------	---------------------------------

Констатации по критичност



■ Critical 2 ■ High 3 ■ Medium 2



Числата са за констатациите, повдигнати в този обобщен примерен документ. Действителен ангажимент със сходен размер обикновено би носил и опашка от хигиенни наблюдения Low и Informational; тук те са пропуснати, за да остане примерът четим. Критичността е оценка по бизнес риск, обяснена в раздел 4.

РЕЗЮМЕ ЗА РЪКОВОДСТВОТО · ПРОДЪЛЖЕНИЕ

Какво означава и какво да направите първо

Сериозните констатации имат една обща тема: данните за достъп и правата не се държат там, където им е мястото. Парола беше разпространена вътре в публично мобилно приложение. Акаунтът на приложението към базата данни има пълен контрол над сървъра на базата данни. Ключът, който доказва кой е влезлият потребител, може да бъде прочетен от всичко, което работи на хоста на приложението. Акаунт с административно ниво във вътрешната директория има парола, която не е сменяна от години и може да бъде поискана от всеки служител.

Нито една от тях не изискваше новаторска атака, специализирано оборудване или неоповестена уязвимост. Всяка е резултат от решение за конфигурация или проектиране, което е било разумно по онова време и никога не е било преразгледано. Това е добрата новина: същите качества, които ги правят лесни за използване, ги правят и прости за поправяне, а по-голямата част от работата е конфигурация, а не разработка.

Рискът освен това е натрупващ се, а не просто сумарен. Сериозните констатации се навързват: данните за достъп от мобилния пакет достигат API-то; акаунтът към базата данни превръща всяка опорна точка в слоя на приложението в контрол над данните на всяка клиентска организация; ключът за подписване превръща достъпа до хоста във възможност да се представиш за администратор. Именно затова общата оценка е High, въпреки че нито една отделна констатация не е пълно проникване сама по себе си.

Трите неща, които да поправите първо

- 1 Подменете всички разкрити данни за достъп, днес. Служебният акаунт и API ключът, открити в мобилното приложение, ключът за подписване на сесии на хоста на приложението и паролата на привилегирования акаунт в директорията трябва да се третират като вече известни на нападател. Подмяната не изисква промяна в кода и издание, а затваря прозореца, докато по-дългите поправки се планират. (F-01, F-03, F-04)

- 2 Свалете акаунта на приложението към базата данни до минимално необходимите права. В момента той има пълен контрол над сървъра на базата данни, включително възможността да изпълнява команди на операционната система. Нужни му са права да чете и записва собствените си данни и нищо друго — една промяна в конфигурацията, която премахва най-големия усилвател в средата. (F-02)
- 3 Прехвърлете обрънатия към интернет шлюз на поддържана платформа. Той вече не получава обновления за сигурност, така че следващата уязвимост в него няма да има налична поправка. Тъй като именно това устройство проверява данните за достъп на служителите, проникване там е незабавен вътрешен проблем. Определете дата и отговорник. (F-05)

Какво следва. Вашите инженери работят от раздел 6, който дава на всяка констатация път за възпроизвеждане, доказателството зад нея и конкретна поправка. Преминаваме през отчета с вашия екип в среща за представяне на резултатите. Когато сте готови, включеният повторен тест проверява всяка поправка и преиздаваме отчета с обновени статуси и писмо за приключване, което можете да споделите с клиенти и одитори.

Нищо тук не изискваше изобретателна атака. Всичко тук изискваше решение, което никога не е било преразгледано — затова по-голямата част от списъка с поправки е конфигурация, не разработка.

РЕЗЮМЕ ЗА РЪКОВОДСТВОТО · НАКРАТКО

РАЗДЕЛ 2

Обхват на ангажимента

Констатация може да съществува само за нещо, което е било в обхвата, достижимо и тествано. Всичко извън тези граници е нетествано, а не сигурно.

Клиент	[Клиент]
Цели	Съставен обхват за този примерен документ: обрънато към клиенти уеб приложение и неговото REST API; публикуваният пакет на мобилното приложение; слой между приложението и базата данни; сегмент от вътрешната мрежа с неговия домейн в Active Directory; и външният периметър. В действителен ангажимент всяка цел е изброена поединично по име на хост, URL, идентификатор на пакет или диапазон от адреси.
Вид на оценката	Grey box, с удостоверяване. Тестови акаунти с няколко нива на права позволиха да бъдат задействани границите на оторизацията между ролите, наред с тестване без удостоверяване на всичко достижимо без акаунт.
Среда	Производствена и тестова среда, както е договорено писмено. Всичко, носещо риск от прекъсване, беше изпълнено първо срещу тестовата среда и срещу производствената само с изрично одобрение.
Период на тестване	[Договорени дати] · работно време, с лица за връзка при спешност на разположение от двете страни през целия период.

Стандарти	OWASP Web Security Testing Guide, OWASP API Security Top 10, OWASP MASVS за мобилния компонент, PTES за работата по мрежата и Active Directory, и CVSS v3.1 за оценяването.
Изключения	Отказ на услуга и тестване под натиск; физическо проникване; социално инженерство срещу служители на клиента; и всяка система, която не е назована в подписания обхват. Вижте Приложение Б.
Екип	Двама старши консултанти, с трети, който извършва независимо осигуряване на качеството и възпроизвежда всяка констатация преди издаването.

Допускания и ограничения

Всеки тест за проникване е моментна снимка, направена при ограничения, а тези ограничения определят какво може да подкрепи резултатът. Те са заявени ясно, за да не прочете никой в този документ повече увереност, отколкото той носи.

- 1 Резултатът е ограничен във времето. Той отразява системите такива, какви бяха през периода на тестване. Внедряване, промяна в конфигурацията или новооповестена уязвимост след този период са извън това, което беше тествано.
- 2 Обхватът е граница, не заключение. Системите, изключени от обхвата, не бяха тествани и нищо тук не бива да се чете като увереност относно тях.
- 3 Липсата на констатация не е доказателство за липса. Никоя оценка с ограничена продължителност не може да докаже, че не е останала уязвимост.
- 4 Предоставената информация беше приета както е дадена. Документацията и данните за достъп, предоставени от [Клиент], бяха използвани както са предадени, без независим одит.
- 5 Наличността беше защитена. Тестове, носещи риск от прекъсване, бяха пропуснати или изпълнени срещу тестова среда, така че някои пътища на атака бяха потвърдени чрез наблюдение, а не чрез изпълнение. Там, където това се отнася, констатацията го посочва.

РАЗДЕЛ 3

Методология

Тестването следваше стандартния ни подход от пет етапа. Всеки етап произвежда определен резултат и ангажиментът не продължава напред, докато този резултат не съществува и не е бил съгласуван.



1 · Обхват и модел на заплахите

Преди да бъде изпратен каквото и да е трафик, уговаряме писмено какво е в обхвата, какво е изрично изключено, кога се провежда тестването и на кого да се обадим, ако нещо се случува. След това изграждаме модел на заплахите: кой реалистично би атакувал тази система, какво би искал и по кои пътища би стигнал до него. Този модел определя къде отиват часовете. Без него тестът се превръща в контролен списък, а контролният списък намира дефектите от контролния списък.

Произвежда: подписан документ за обхвата и правилата на ангажимента, посочено лице за контакт от двете страни и приоритизиран списък със сценарии за атака.

2 · Разузнаване и картографиране

Изброяваме действителната повърхност за атака, а не документираната: хостове, услуги, маршрути, параметри, роли, процеси на удостоверяване, пакети от страна на клиента, зависимости от трети страни и всичко, което организацията е публикувала, без да иска. Автоматизирани инструменти се използват тук за покритие и бързина, защото широчината е задача за машина.

Произвежда: пълна карта на тестваната повърхност, включително всичко в обхвата, за което клиентът не е знаел, че е изложено.

3 · Ръчно използване на уязвимости

Тук ангажиментът се печели или губи и това е изцяло човешка работа. Всеки проблем-кандидат се тества на ръка, навързва се с други, където навързването е възможно, и — където е безопасно — се използва достатъчно далеч, за да бъде доказано действително въздействие, а не теоретичен риск. Границите на оторизацията се задействат с множество акаунти. Бизнес логиката се атакува по собствените ѝ правила. Нищо не достига отчета на основание на твърдението на един инструмент.

Произвежда: проверени констатации с доказателство, път за възпроизвеждане и демонстрирано въздействие.

4 · Отчитане

Всяка констатация се описва с доказателството си, стъпките за възпроизвеждането ѝ, какво означава за бизнеса и конкретна поправка, която подхожда на технологията пред нас. Втори старши консултант преглежда отчета и независимо възпроизвежда всяка констатация, преди той да бъде издаден. След това преминаваме през отчета с екипа на клиента в среща за представяне на резултатите.

Произвежда: този документ, машинно четим експорт на констатациите за системата за проследяване на клиента и сесия за разговор по резултатите.

5 · Повторен тест

Щом отстраняването е завършено, тестваме повторно всяка констатация, потвърждаваме какво действително е поправено и преиздаваме отчета с обновени статуси и подписано писмо за приключване. Това е включено по стандарт, а не се продава отделно, защото непроверена поправка е план, а не резултат.

Произвежда: обновен отчет със статус за всяка констатация и писмо за приключване, подходящо за клиенти и одитори.

МЕТОДОЛОГИЯ · ПРОДЪЛЖЕНИЕ

Инструменти, проверка и безопасност

Как се използват инструментите

Използваме автоматизирани инструменти интензивно и им се доверяваме за точно едно нещо: покритие. Скенерът може да изброи голяма среда по-бързо и по-последователно от човек и няма да му стане скучно на четиристотния хост. Това, което не може, е да разбере за какво служи приложението. Той не знае кои записи един акаунт има право да вижда, кои стъпки от работния процес могат да бъдат прескочени, или кой от два отговора с правилен вид представлява прекосена граница.

Затова разделението на труда е ясно. Автоматизацията картографира повърхността и повдига кандидати. Човек решава кое е действително. Всяка констатация в този отчет беше проверена ръчно и, където беше безопасно, използвана, за да се демонстрира действително въздействие. Нищо не се отчита на основание на банер с версия или на оценка за увереност от инструмент, а кандидати, които се окажат неизползваеми в контекста, се отхвърлят, вместо да бъдат добавени за обем в списъка с констатации.

Какво добавя автоматизацията

- Откриване на хостове, портове и услуги в целия обхват.
- Обхождане и изброяване на параметри по повърхността на приложението.
- Разпознаване на версии и конфигурации в голям обем.
- Кандидати за известни уязвимости, за човешки триаж.
- Покритие за регресии при повторния тест.

Какво добавя само човекът

- Тестване на оторизацията през множество акаунти и роли.
- Злоупотреба с бизнес логиката, която произвежда отговор с валиден вид.
- Навързване на проблеми с ниска критичност в сериозен изход.
- Преценка на действителното въздействие в тази среда, не абстрактно.
- Решение какво е безопасно да бъде използвано в производствена система.

Безопасност и правила на ангажмента

Тестването на производствена система носи действителен риск, а контролите върху този риск се договарят писмено, преди да бъде изпратена първата заявка. Правилата по-долу се прилагаха през цялото време.

- 1 Разрешението е първо. Никакво тестване не започва, докато не е налице подписано разрешение, назоваващо системите в обхвата, и екипът го носи през цялото време.

- 2 Наличността е защитена. Отказът на услуга и тестването под натиск са изключени по подразбиране. Всичко с правдоподобен път до прекъсване се изпълнява срещу тестовата среда или срещу производствената само с конкретно одобрение в момента.
- 3 Минимално необходимо доказателство. Използването на уязвимостта спира в момента, в който въздействието е демонстрирано. Не извличаме данни масово, не унищожаваме и не променяме записи и не се задържаме в средата повече от необходимото за доказателството.
- 4 Жив контакт през цялото време. Посочените лица за контакт от двете страни са достъпни през целия период. Предполагам действителен инцидент, вече съществуващо проникване или неочаквано прекъсване се докладват незабавно, а не в края.
- 5 Критичните констатации се ескалират при откриване — в същия ден, в който бъдат потвърдени, със съвет за междинно ограничаване, вместо да бъдат задържани за отчета.
- 6 Чисто излизане. Тестовите акаунти, качените файлове и всеки друг артефакт, създаден по време на тестването, се премахват и изброяват, за да може клиентът да провери, че средата е както е била.

РАЗДЕЛ 4

Оценки на критичността

Констатациите се оценяват по бизнес риск — какво струва проблемът на тази организация, ако бъде използван — като CVSS v3.1 информира оценката, но не я диктува.

Оценка	Какво означава	Очакван отговор
CRITICAL	Незабавно и тежко бизнес въздействие; лесна експлоатация. Обикновено без удостоверяване и надеждно, водещо директно до загуба на данни, финансова загуба или контрол над средата.	Поправете веднага. Ескалирайте същия ден; приложете междинно ограничаване, ако пълната поправка отнема време.
HIGH	Значително въздействие и реалистична възможност за експлоатация, макар да може да изисква позиция, акаунт или предварително условие. Често решаващото звено във верига.	Поправете в текущия цикъл. Не допускайте да влезе в общия списък със задачи.
MEDIUM	Умерено въздействие или сериозно въздействие, което зависи от условия извън пълния контрол на нападателя.	Планирайте отстраняване с договорен срок в следващия цикъл или два.
LOW	Ограничено пряко въздействие. Укрепване на защитата и защита в дълбочина; полезно за нападател главно в комбинация с нещо друго.	Планирайте го наред със свързаната работа.
INFORMATIONAL	Няма пряко въздействие върху сигурността. Наблюдение или отклонение от добрата практика.	Отбележете го. Действайте по ваша преценка.

Как се използва CVSS — и къде спира

Всяка констатация тук носи базова оценка и вектор по CVSS v3.1. CVSS е общ език: той позволява на одитор, доставчик и инженер да сравнят две констатации, без да спорят за прилагателни.

Това, което CVSS умишлено не знае, е вашият бизнес. Базовата оценка описва уязвимостта абстрактно: тя няма мнение дали засегнатата система съдържа регулирани лични данни или тестово копие на миналогодишния каталог, дали компенсиращ контрол прави атаката непрактична, или дали прекъсване там спира търговията.

Затова използваме базовата оценка като входни данни и оценяваме по бизнес риск. Там, където двете се разминават, констатацията посочва оценката, резултата и причината. Широко известна уязвимост, която не е използвана във вашия контекст, ще бъде оценена тук като Low; скромна на вид проблем по пътя към най-чувствителните ви данни ще бъде оценен като High.

Оценките в този примерен документ са илюстративни. Този документ е съставен от анонимизирани ангажименти със синтетични данни, така че показаните оценки и вектори по CVSS са представителни примери, подбрани да съответстват на всеки вид констатация. Те не са оценки, присвоени на действителна система, и не бива да бъдат цитирани като такива.

CVSS ви казва колко лоша е уязвимостта. Само вашият контекст ви казва колко лош ще бъде вторникът.

РАЗДЕЛ 4 · НАКРАТКО

РАЗДЕЛ 5

Обобщение на констатациите

Всичките седем констатации, в реда на отстраняване. Раздел 6 излага доказателството, пътя за възпроизвеждане, въздействието и препоръката за всяка от тях.

ID	Констатация	Компонент	Критичност	CVSS	Статус
F-01	Hardcoded backend credentials in a shipped mobile app	Mobile app bundle	CRITICAL	9.1	Отворена
F-02	Application database account is a cluster superuser	App-to-database connection	CRITICAL	9.1	Отворена
F-03	Production session-signing key recoverable from the app host	Web session / token signing	HIGH	8.1	Отворена
F-04	Kerberoastable Domain Admin service account, password unchanged for years	Active Directory service account	HIGH	8.1	Отворена
F-05	Internet-facing remote-access gateway on an end-of-life OS	External authentication gateway	HIGH	8.1	Отворена
F-06	Long-lived API token leaked via URL fragment and JavaScript global	SSO handoff / single-page app	MEDIUM	6.4	Отворена
F-07	Stored spreadsheet formula injection in data export	Table / CSV export	MEDIUM	6.4	Отворена

Статусът е към датата на този отчет. След повторния тест всеки ред се произвежда като затворен, частично отстранен или отворен, с бележка, обясняваща всеки статус, различен от затворен. Базовите оценки по CVSS v3.1 са илюстративни в този примерен документ.

Как се чете таблицата

Констатациите са подредени по критичност, която тук е и редът, в който бихме отстранявали. Това не е винаги така: там, където констатация Medium е предпоставка за Critical, или където една промяна в конфигурацията затваря няколко реда наведнъж, разговорът по резултатите ще го посочи.

- 1 Пет от седем констатации са за данни за достъп или права. F-01 до F-04 и F-06 са, в основата си, тайна, държана на погрешно място, или акаунт с повече права, отколкото работата му изисква — един организационен проблем в пет системи, най-добре третиран като една програма от работа, а не като пет задачи.
- 2 Нито една констатация не изискваше новаторска атака. Няма zero-day, няма екзотична техника. Всяка констатация зависи от решение за конфигурация или проектиране, така че поправките са предимно експлоатационни и могат да започнат преди следващото издание.
- 3 Двете констатации Medium не са запълване. F-06 произвежда данни за достъп с многогодишна валидност, които надживяват изхода от системата; F-07 поставя изпълнимо съдържание на работната станция на вътрешен проверяващ. И двете са Medium, защото всяка изисква предварително условие, а не защото резултатът е малък.

Предава се и. Наред с този PDF получавате същите констатации като машинно четим експорт в CSV и JSON, така че идентификаторите, критичностите и статусите се зареждат директно във вашата система за проследяване на задачи.

РАЗДЕЛ 6

Подробни констатации

По един раздел за всяка констатация: критичност и илюстративен CVSS, засегнат компонент, какво беше открито, доказателството, път за възпроизвеждане, въздействието и поправка.

За доказателствата по-долу. В действителен отчет всеки запис е заличена снимка на екрана от собствената сесия на инженера, с пълния текст в приложението. Всеки запис тук е синтетична реконструкция — командите са представителни; хостовете, акаунтите, ключовете и изходът са измислени.

CRITICAL

F-01 Hardcoded backend credentials in a shipped mobile app

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Critical	9.1 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Пакет на мобилното приложение	Отворена

Class: Use of hard-coded credentials (CWE-798) · Reference: OWASP MASVS-STORAGE-1 · OWASP API Top 10 API2:2023 · Composite element: mobile application assessment

Описание

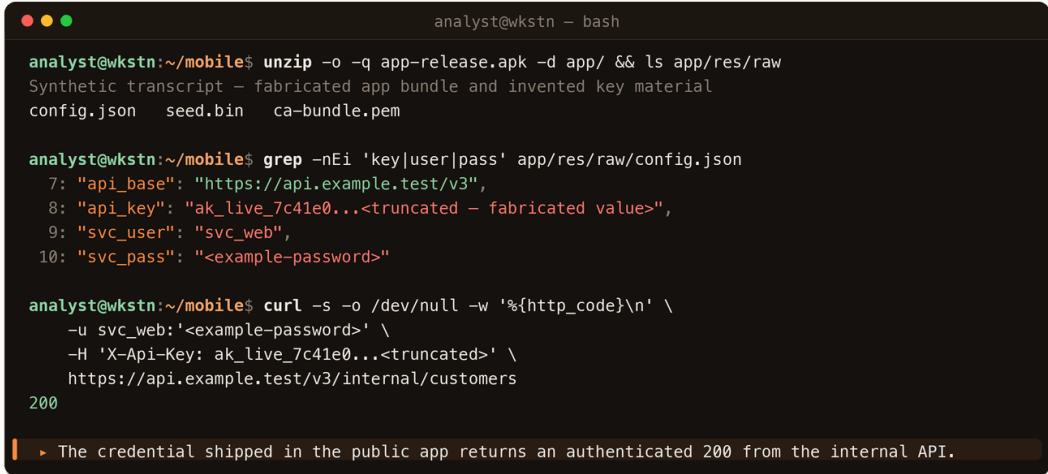
The published application bundle was downloaded from its distribution channel, unpacked and searched for embedded secrets. A configuration asset inside it carried a service account name, its password and a long-lived API key in plaintext.

The credentials were not inert. Replayed from a workstation with no connection to the mobile estate, they returned an authenticated response from an internal customer-data API that is not otherwise reachable without a valid session. The application was not holding a credential for its own limited use — it was publishing a working service identity to everyone who installs it, in every build that carried the same asset.

Доказателство

Backend credentials hardcoded in the shipped mobile appSYNTHETIC EXAMPLE

Anyone who installs the published bundle obtains a working service credential.



```
analyst@wkstn ~$ unzip -o -q app-release.apk -d app/ && ls app/res/raw
Synthetic transcript - fabricated app bundle and invented key material
config.json  seed.bin  ca-bundle.pem

analyst@wkstn ~$ grep -nEi 'key|user|pass' app/res/raw/config.json
7: "api_base": "https://api.example.test/v3",
8: "api_key": "ak_live_7c41e0...<truncated - fabricated value>",
9: "svc_user": "svc_web",
10: "svc_pass": "<example-password>"

analyst@wkstn ~$ curl -s -o /dev/null -w '%{http_code}\n' \
-u svc_web:'<example-password>' \
-H 'X-API-Key: ak_live_7c41e0...<truncated>' \
https://api.example.test/v3/internal/customers
200

▶ The credential shipped in the public app returns an authenticated 200 from the internal API.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-01 — the bundle is unpacked, the embedded service credential and API key are read from a configuration asset, and the pair returns an authenticated 200 from the internal API. Illustrative reconstruction; synthetic data, not client evidence.

Стъпки за възпроизвеждане

- 1 Obtain the published application bundle from its distribution channel and unpack it to a working directory.
- 2 Search the bundled resources for credential-shaped strings; in the sample transcript **grep -nEi 'key|user|pass'** against the configuration asset returns the API base URL, the API key and the service account name and password.
- 3 Replay the recovered pair against the API base URL from an unrelated host, with no device, no app and no interactive login.
- 4 Observe an authenticated **200** from the internal customer endpoint, confirming the credential is live rather than a build-time placeholder.

Въздействие

Anyone who installs the published application obtains a working service identity against an internal customer-data API: unauthenticated access to regulated personal data at internet scale, with no account to disable and nothing anomalous for a login-monitoring control to notice, because the requests are correctly authenticated — just not by the intended party. The same key also widens the blast radius of any future application-tier bug, since an attacker starts with a valid identity rather than having to earn one.

Препоръка

- 1 Treat the exposed service account and API key as compromised and rotate both immediately, before any code change ships.

- 2 Remove every embedded credential from the bundle and from version control history; a secret that was once committed is not removed by deleting the line.
- 3 Broker authentication server-side: the application should obtain a short-lived, per-user token through SSO or an OAuth authorization-code flow with PKCE, and should never hold a shared service identity.
- 4 Add secret scanning to the build pipeline and fail the build on a positive, so the next instance is caught before release rather than at the next test.

CRITICAL

F-02 Application database account is a cluster superuser

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Critical	9.1 AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H	Връзка приложение–база данни	Отворена

Class: Execution with unnecessary privileges (CWE-250) · Reference: OWASP ASVS V1.4 · CIS Database Benchmark · Composite element: web application and infrastructure assessment

Описание

The account the application uses to reach its database holds the cluster superuser attribute. Connecting with the application's own credentials and querying the role catalogue confirmed it directly; listing databases showed the same role owning every tenant schema on the instance, not only the one the application serves. Superuser is not simply a wide grant — on this engine it also unlocks server-side command execution, so the database account is an operating-system foothold wearing a SQL interface.

Nothing about this is exploitable on its own. It needs an application-tier flaw, a leaked connection string or access to an application host first, and that is precisely why it matters: it is the amplifier that converts any one of those ordinary events into a full-cluster and full-host compromise, and it was present on every path we examined.

Доказателство

The application database account is a cluster superuser

One application-tier flaw becomes full control of every tenant database on the cluster.

SYNTHETIC EXAMPLE

```
analyst@wkstn ~$ psql "host=db-01.example.test user=app_rw dbname=appdb"
Synthetic transcript — invented cluster, roles and output
appdb=# SELECT current_user, usesuper FROM pg_user WHERE username = current_user;
current_user | usesuper
-----
app_rw       | t

appdb=# \l -- databases visible to this role
Name | Owner | Access privileges
-----
appdb | app_rw |
tenant_a | app_rw |
tenant_b | app_rw |

appdb=# COPY (SELECT 1) TO PROGRAM 'id -un';
COPY 1 -- superuser-only: OS command runs as the DB service user
> usesuper = t: the app role reads every tenant database and runs OS commands on the host.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-02 — the application role reports **usesuper = t**, every tenant database is visible to it, and a superuser-only statement executes an operating-system command on the database host. Illustrative reconstruction using synthetic data on reserved example.test names; not client evidence.

Стъпки за възпроизвеждане

- 1 Connect to the database using the application's own connection parameters (host, role and database taken from the application configuration).
- 2 Query the role catalogue for the current user's superuser attribute; the sample transcript shows **usesuper** returning **t**.
- 3 List the databases visible to the role and confirm that tenant schemas belonging to other customers are owned by, and readable to, the same account.
- 4 Issue a superuser-only statement that reaches the operating system — in the sample, **COPY (SELECT 1) TO PROGRAM 'id -un'** — and observe it succeed.

Въздействие

Any compromise of the application process — an injection flaw, a leaked environment file, a stolen deployment artefact, a mis-scoped backup — becomes read and write access to every tenant's data and command execution on the database host. For a multi-tenant platform that is the difference between one customer's incident and every customer's incident, and it converts a contained application bug into a reportable breach. It also removes the audit value of database logging, since the account can alter the records that would describe the event.

Препоръка

- 1 Demote the application role now. It needs **SELECT / INSERT / UPDATE / DELETE** on its own schema and nothing more — no superuser, no ownership of other tenants' objects, no object creation at runtime.
- 2 Separate duties: a migration role with schema rights used only by the deployment pipeline, and a runtime role with data rights used only by the application.
- 3 Restrict the database at the network layer so only application hosts can reach the listener, and require TLS with certificate verification.
- 4 Add a recurring check that fails if any application role gains superuser, **CREATEROLE** or ownership outside its own schema.

HIGH

F-03 Production session-signing key recoverable from the app host

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
High	8.1 AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N	Подписване на уеб сесии / токени	Отворена

Class: Insufficiently protected credentials (CWE-522, CWE-321) · Reference: OWASP ASVS V3.5 · OWASP WSTG-SESS · Composite element: web application and internal infrastructure assessment

Описание

The secret used to sign production session tokens was held in a world-readable configuration file on the application host. Any local account — an unprivileged service account, a build agent, a monitoring collector, or anything that lands through an unrelated local flaw — can read it.

Possession of a signing key is possession of the authentication system. With the key in hand a valid session can be minted offline for any subject, any role and any tenant, with an arbitrary lifetime. The forged token never touches the login flow, so multi-factor authentication, lockout thresholds, conditional access and sign-in logging are bypassed rather than defeated: there is no authentication event to evaluate or record. We verified the chain

end to end against an administrative endpoint, which returned the expected administrative response.

Доказателство

Production session-signing key readable from the app host

A world-readable key file lets any local account mint a valid session for any user.

SYNTHETIC EXAMPLE

```
analyst@app-01 ~$ ls -l /etc/appsvc/secrets.yml
Synthetic transcript - invented host, file and key material
-rw-r--r-- 1 appsvc appsvc 412 Mar 11 09:22 /etc/appsvc/secrets.yml

analyst@app-01 ~$ grep session_secret /etc/appsvc/secrets.yml
session_secret: "s3cr3t_9f14ab07...<truncated - fabricated value>"

analyst@app-01 ~$ python3 forge_session.py --sub bob --role admin --ttl 30d
signed offline with the recovered key - no interaction with the login flow
eyJhbGciOiI...<truncated>.eyJzdWIiOiJib2IiLCJyb2x1IjoieWRTaW4i...<truncated>.Zm9yZ2Vh...<truncated>

analyst@app-01 ~$ curl -s -H "Cookie: session=$FORGED" \
https://app.example.test/admin/users | head -1
HTTP/1.1 200 OK
{"user": "bob", "role": "admin"}

▶ The server accepts an offline-forged token and grants bob the admin role.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-03 — a world-readable key file yields the production session secret, a token is signed offline for an arbitrary subject and role, and the server accepts it. Illustrative reconstruction using synthetic data on reserved example.test names and the placeholder users alice / bob; not client evidence.

Стъпки за възпроизвеждане

- 1 From any local account on the application host, list the permissions on the application secrets file; the sample transcript shows mode `-rw-r--r--`, readable by every local user.
- 2 Read the session-signing secret out of that file.
- 3 Sign a token offline for an arbitrary subject and role — in the sample, subject **bob** with the administrative role and a 30-day lifetime — with no interaction with the login flow.
- 4 Present the forged token as the session cookie against a privileged endpoint and observe **200 OK** with the elevated role reflected in the response body.

Въздействие

Authentication can be bypassed entirely for any user, including administrators, without a password, without a second factor and without generating a sign-in record. Account takeover becomes silent and unbounded, and any action attributed to a user in the application logs stops being evidence of what that user did — which undermines both incident response and any audit or non-repudiation claim built on those logs. Rotating the key invalidates every live session, so the cost of the fix grows the longer it is deferred.

Препоръка

- 1 Rotate the signing key immediately and invalidate all existing sessions. Plan the forced re-authentication; do not let the disruption delay the rotation.
- 2 Move the key out of the filesystem into a managed secrets store or an HSM, fetched at process start and never written to disk or to a log. If a file must remain, restrict it to the service account with mode **0600**, enforced by configuration management.

- 3 Adopt asymmetric signing so verifying services hold only the public key and a compromised verifier cannot mint tokens.
- 4 Introduce key identifiers, a scheduled rotation and server-side revocation of individual sessions, so rotating becomes routine rather than an incident.

HIGH

F-04 Kerberoastable Domain Admin service account, password unchanged for years

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
High	8.1 AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N	Active Directory service account	Отворена

Class: Insufficiently protected credentials (CWE-522) · ATT&CK: T1558.003 · Reference: CIS Microsoft Windows Benchmarks · Microsoft AD tiering guidance · Composite element: internal network and Active Directory assessment

Описание

A service account with a registered service principal name is a direct member of the Domain Admins group, and its password had not been changed in several years. Any account that can authenticate to the domain — the lowest-privileged ordinary user, a shared kiosk login, a contractor, a compromised workstation — can request a service ticket for that principal. The ticket is encrypted with a key derived from the account's password and cracks offline, at any speed, with no further contact with the environment.

This is not a flaw in Kerberos; the protocol behaves exactly as designed. What makes it decisive is the combination of three choices: a human-chosen password, no rotation, and membership of the most privileged group in the directory. In the sample transcript the ticket recovers against an ordinary wordlist and the credential then enumerates Domain Admins — which is the whole directory.

Доказателство

Kerberoastable service account is a Domain Admin

SYNTHETIC EXAMPLE

A requestable service ticket cracks offline to an account with full domain privileges.

```
operator@jump-01 ~$ GetUserSPNs.py EXAMPLE.TEST/alice -dc-ip 192.0.2.10 -request
Synthetic transcript — invented SPN, fabricated ticket and crack result
ServicePrincipalName      Name      MemberOf
-----
MSSQLSvc/db-01.example.test  svc_sql   CN=Domain Admins,CN=Users

$krb5tgs$23$*svc_sql$EXAMPLE.TEST$MSSQLSvc/db-01*$
a17f3c0b9e2d4a8f...<truncated — fabricated value>

operator@jump-01 ~$ hashcat -m 13100 tgs.txt wordlist.txt
Recovered.....: 1/1  svc_sql:<example-password>

operator@jump-01 ~$ net rpc group members "Domain Admins" -U svc_sql
EXAMPLE\administrator
EXAMPLE\svc_sql

> svc_sql cracks offline and is a member of Domain Admins — full domain compromise.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-04 — an ordinary domain account requests a service ticket for a principal that is a member of Domain Admins, the ticket cracks offline, and the recovered credential authenticates with full domain privilege. Illustrative reconstruction using synthetic data, reserved EXAMPLE.TEST names, the TEST-NET address 192.0.2.10 and the placeholder accounts alice / svc_sql; not client evidence.

Стъпки за възпроизвеждане

- 1 From any authenticated domain account, enumerate accounts that have a service principal name registered and note their group memberships.
- 2 Identify a principal that is a member of a privileged group — in the sample, **svc_sql** in **Domain Admins**.
- 3 Request a service ticket for that principal and export it in a crackable format. This is an ordinary Kerberos request and is logged as one.
- 4 Crack the ticket offline against a wordlist, then authenticate with the recovered credential and enumerate a privileged group to confirm the access obtained.

Въздействие

Any authenticated user in the domain can obtain Domain Admin. From there every domain-joined system, every file share, every backup target and every credential held in the directory is reachable, and the attacker can create persistence that survives password resets and rebuilds. This is the point at which an incident stops being about one host and starts being about the entire estate, and the offline crack leaves nothing anomalous in the authentication logs.

Препоръка

- 1 Remove the account from Domain Admins. A service that genuinely needs that level of privilege is almost always a service that needs redesigning.
- 2 Convert it to a group-managed service account so the password is machine-generated and rotated automatically; where that is not possible, set a random passphrase of at least 25 characters and rotate it on a schedule.
- 3 Audit every account in the directory carrying a service principal name and treat privileged ones as a standing exception requiring sign-off. Apply tiering so privileged credentials never authenticate to workstations or ordinary members.
- 4 Alert on service-ticket requests for privileged principals using weak encryption types, and on any bulk enumeration of service principal names.

HIGH

F-05 Internet-facing remote-access gateway on an end-of-life OS

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
High	8.1 AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	External authentication gateway	Отворена

Class: Use of unmaintained components (CWE-1104) · information exposure (CWE-200) · Reference: OWASP WSTG-INFO · PCI DSS 6.3.3 · CIS Controls 7 · Composite element: external perimeter assessment

Описание

The remote-access gateway that fronts staff authentication is published to the internet and runs an operating system build that left vendor support some time ago. Unsupported means no security patches are issued at all — the exposure is not a list of known missing fixes but an open-ended commitment to accept every future vulnerability in the platform permanently. The same host also discloses internal naming: the response headers carry the backend host name and an unauthenticated diagnostic endpoint returns the internal domain and node identifier.

Individually those are low-severity information leaks. Here they sit on the device that terminates authentication for the internal directory, so they hand an attacker the naming convention, the domain and a confirmed target before any credential is attempted. We did not attempt to exploit a platform vulnerability against this host: exploitation of an unsupported production gateway carries a real risk of outage, and the rules of engagement excluded denial of service. The finding is evidenced by version and build confirmation rather than by a shell.

Доказателство

Internet-facing remote-access gateway runs an end-of-life OS

The gateway exposes an out-of-support build and leaks internal naming in its response.

SYNTHETIC EXAMPLE

```
analyst@wkstn ~$ nmap -sV -p 443 --script http-title vpn.example.test
Synthetic transcript - invented gateway, banner and build string
Nmap scan report for vpn.example.test (192.0.2.80)
PORT      STATE SERVICE VERSION
443/tcp   open  https    gateway httpd (build 8.1-eol)

analyst@wkstn ~$ curl -skI https://vpn.example.test/ | grep -iE 'server|x-'
Server: Gateway-OS 8.1 (unsupported since 2023)
X-Backend-Host: gw-01.corp.example.test

analyst@wkstn ~$ curl -sk https://vpn.example.test/diag | head -2
{"domain":"corp.example.test", "node":"gw-01"}

► An internet-facing gateway on an end-of-life OS, also leaking the internal domain and host name.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-05 — service and build identification against the internet-facing gateway, the unsupported platform string in the response headers, and an unauthenticated diagnostic endpoint returning the internal domain and node name. Illustrative reconstruction using synthetic data on reserved example.test names and the TEST-NET address 192.0.2.80; not client evidence.

Стъпки за възпроизвеждане

- 1 Identify the service and build on the published gateway with a version scan against the TLS service.
- 2 Request the root resource and inspect the response headers; the sample shows the platform string with its unsupported build and an **X-Backend-Host** header carrying the internal host name.
- 3 Request the unauthenticated diagnostic endpoint and observe the internal domain and node identifier in the response body.
- 4 Cross-reference the identified build against the vendor's support lifecycle to confirm it no longer receives security updates. No exploitation was attempted against the live gateway.

Въздействие

An authentication boundary that cannot be patched will eventually be breached by a vulnerability nobody has published yet, and the organisation will have no remedy available on the day it is. Because this device terminates authentication for the internal directory, a compromise there is not a perimeter incident — it is a credential incident with an internal foothold attached. The disclosed internal names shorten the reconnaissance phase of exactly that attack.

Препоръка

- 1 Migrate the gateway role to a supported, currently patched platform. Treat this as a dated project with an owner, not a backlog item.

- 2 Until migration completes, restrict administrative interfaces to known source ranges, enforce multi-factor authentication for every account that can traverse the gateway, and raise monitoring on it.
- 3 Suppress internal naming in responses: remove the backend host header, and require authentication on — or remove — the diagnostic endpoint.
- 4 Add the asset to a lifecycle inventory that records vendor end-of-support dates and raises them before they pass, then re-scan the external estate for other unsupported builds.

MEDIUM

F-06 Long-lived API token leaked via URL fragment and JavaScript global

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Medium	6.4 AV:N/AC:H/PR:L/UI:R/S:U/C:H/I:H/A:N	Предаване при SSO / едностранично приложение	Отворена

Class: Sensitive information in a URL (CWE-598) · insufficient session expiration (CWE-613) · Reference: OWASP ASVS V3.3 · OAuth 2.0 Security BCP (RFC 9700) · Composite element: web application and API assessment

Описание

The single-page application receives its API access token from the SSO handoff in the URL fragment and then assigns it to a global variable on the page. Both are poor places for a bearer credential: the fragment is written into browser history and session-restore data and is visible to anything with access to the address bar or the profile on disk, while the global is readable by every script running in the page, including any third-party tag or analytics bundle.

The token's lifetime makes the exposure materially worse. The decoded claims show a validity measured in years rather than minutes, and the token continued to authenticate after the user had logged out: logout clears the client state but does not revoke the credential server-side. The rating is Medium rather than High only because a precondition is needed — the browser, the history, a shared workstation or a script in the page.

Доказателство**Long-lived API token leaked via URL fragment and JS global**

SYNTHETIC EXAMPLE

A token with a multi-year lifetime is exposed client-side and survives logout.

```
analyst@wkstn:~$ cat captured-url.txt
Synthetic transcript — invented token and decoded claims
https://app.example.test/dashboard#access_token=eyJhbGciOi...<truncated>

analyst@wkstn:~$ node -e "console.log(window.__API_TOKEN)" # page global
eyJhbGciOi...<truncated>

analyst@wkstn:~$ node decode.js --field exp
iat: 2026-01-04 exp: 2029-01-03 (lifetime ~1095d)

analyst@wkstn:~$ curl -s -o /dev/null -w '%{http_code}\n' \
-H "Authorization: Bearer eyJhbGciOi...<truncated>" \
https://api.example.test/v3/me # replayed after logout
200
```

► The fragment/JS-exposed token has a ~3-year lifetime and still authenticates after logout.

i Illustrative example — synthetic data, not client evidence.

Evidence for F-06 — the access token arrives in the URL fragment, is also readable from a page global, decodes to a multi-year lifetime, and still authenticates to the API after logout. Illustrative reconstruction using synthetic data on reserved example.test names; not client evidence.

Стъпки за възпроизвеждане

- 1 Complete the SSO handoff and capture the resulting URL; the access token is present in the fragment after the `#`. Read the same token from the page's JavaScript global to confirm it is also exposed to any script in the document.
- 2 Decode the token claims and compare the issued-at and expiry values; the sample shows a lifetime of roughly three years.
- 3 Log out of the application through the user interface.
- 4 Replay the captured token as a bearer credential against an API endpoint and observe **200**, confirming no server-side revocation took place.

Въздействие

A token recovered from browser history, a shared or kiosk workstation, a synchronised browser profile, a screen share or a third-party script grants full API access as that user for years, and logging out does not take it away. On a shared device the next person to sit down can retrieve it from history without any technical skill. Because there is no revocation path, the only response to a suspected leak is to rotate the signing key and disrupt every user at once.

Препоръка

- 1 Use the OAuth 2.0 authorization-code flow with PKCE so that no token is ever delivered in a URL fragment.
- 2 Keep tokens out of page globals. Hold them in memory within a closure, or in an HttpOnly, Secure, SameSite cookie issued by a server-side session broker.
- 3 Shorten access-token lifetime to minutes and use refresh tokens that are rotated on each use and bound to the client.
- 4 Revoke tokens server-side on logout, on password change and on role change, and give support staff a way to revoke an individual session.

MEDIUM

F-07 Stored spreadsheet formula injection in data export

КРИТИЧНОСТ	CVSS v3.1 (ИЛЮСТРАТИВНО)	ЗАСЕГНАТ КОМПОНЕНТ	СТАТУС
Medium	6.4 AV:N/AC:H/PR:L/UI:R/S:U/C:H/I:H/A:N	Експорт на таблица / CSV	Отворена

Class: Improper neutralisation of formula elements in a CSV file (CWE-1236) · Reference: OWASP WSTG-INPV · OWASP ASVS V5.3 · Composite element: web application assessment

Описание

A user-controlled profile field accepts a value beginning with an equals sign and stores it verbatim. The application's export feature then writes that value into a CSV file without neutralising it, so the cell is emitted as a live formula rather than as text. When the export is opened in a spreadsheet application, the formula is evaluated on the machine of whoever opened it.

The exposure is routinely misread as cosmetic because the application itself is unaffected. The vulnerability does not execute in the web tier at all — it executes in the desktop application of an internal reviewer in operations, finance or support who opens the export as part of their job. Depending on the spreadsheet client and its configuration, formula payloads can reach external data, trigger a protocol handler or prompt the user to run a command, and the prompt arrives attached to a file the user has every reason to trust. Those conditions are

common rather than guaranteed, which is why the rating is Medium.

Доказателство

Stored formula injection reaches the data exportSYNTHETIC EXAMPLE

A value beginning with = is stored verbatim and exported as a live spreadsheet formula.

```
analyst@wkstn: ~$ curl -s -X POST https://app.example.test/api/v3/profile \
-d 'display_name==2+5+cmd|"/c calc"!A1'
Synthetic transcript - invented record and exported file
{"status":"saved"}

analyst@wkstn: ~$ curl -s https://app.example.test/api/v3/export.csv -o out.csv
analyst@wkstn: ~$ cat out.csv
id,display_name,email
1,alice,alice@example.test
2,=2+5+cmd|"/c calc"!A1,bob@example.test

the stored value reaches the CSV with its leading = intact
# a spreadsheet client parses the cell as a formula, not text,
# so it executes in the reviewer's application

! > The = value reaches the export as a live formula that runs in the reviewer's spreadsheet.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-07 — a profile value beginning with = is stored verbatim and reaches the generated CSV with its leading character intact, so a spreadsheet client parses the cell as a formula. Illustrative reconstruction using synthetic data on reserved example.test names and the placeholder users alice / bob; not client evidence.

Стъпки за възпроизвеждане

- 1 Authenticate as an ordinary low-privilege user and set a profile field to a value whose first character is = and which contains a spreadsheet formula; the application accepts and stores it.
- 2 Trigger the data export that includes that field — typically from an administrative or reporting view used by another member of staff.
- 3 Inspect the generated file and confirm the stored value appears with its leading = intact and is neither quoted nor prefixed.
- 4 Open the file in a spreadsheet client in a controlled test environment and observe the cell being parsed as a formula rather than as text. No payload was executed against any live workstation during this assessment.

Въздействие

An unprivileged user can place executable content into a file that a trusted internal reviewer will open on a corporate workstation — a path from the lowest privilege level in the application to code execution inside the office network, bypassing the perimeter because the file arrives through the organisation's own reporting process. It also poisons exports shared with auditors, regulators and customers.

Препоръка

- 1 Neutralise every exported cell whose first character is =, +, -, @, a tab or a carriage return — prefix with a single quote or wrap the value so the client treats it as text.
- 2 Apply the control in the export layer, once, for every export in the product. Per-field or per-endpoint fixes reliably miss the next export someone adds.
- 3 Where the format allows it, write typed cells and declare the column as text rather than relying on the client's parsing heuristics, and add a regression test that round-trips a formula-shaped value through store and export and asserts it comes back inert.

Отстраняването накратко

Същите седем препоръки, сведени до решението, което всяка изисква. Усилието е нашата оценка за самата работа, не за контрола на промените около нея.

ID	Какво да направите	Кой отговаря за това	Усилие	Кога
F-01	Подменете разпространените данни за достъп и ключ; посредничете за удостоверяването от страна на сървъра	Мобилни и платформа	Часове, после дни	Днес, после в този цикъл
F-02	Свалете ролята на приложението в базата данни до минимално необходимите права	База данни и платформа	Часове	Днес
F-03	Подменете ключа за подписване; преместете го в хранилище за тайни	Платформа	Часове, после дни	Днес, после в този цикъл
F-04	Премахнете акаунта от Domain Admins; превърнете го в управляван служебен акаунт	Active Directory	Часове	Този цикъл
F-05	Мигрирайте ролята на шлюза към поддържана и обновена платформа	Инфраструктура	Седмици (проект)	План с дата сега
F-06	Преминете към authorization code с РКСЕ; съкратете и отменяйте токените	Уеб и идентичност	Дни	Следващият цикъл
F-07	Неутрализирайте символите за формули в слоя за експортиране	Приложение	Часове	Следващият цикъл

Напомняне за този раздел. F-01 до F-07 са взети от няколко различни анонимизирани ангажимента и представени тук, сякаш идват от един. Всеки запис, име на хост, адрес, акаунт и оценка по-горе са синтетични и илюстративни и не се описва нито един действителен клиент.

РАЗДЕЛ 7

Повторен тест и приключване

Отстраняването е завършено само когато някой е атакувал поправката. Повторният тест е включен в цената на ангажимента по стандарт — той не е добавка и не се остойносттава отделно.

Какво покрива повторният тест

Когато екипът ви потвърди, че отстраняването е завършено, се връщаме и тестваме отново всяка констатация в този отчет. Повторният тест не е преглед на документация: всяка констатация се атакува отново по същия път за възпроизвеждане, записан в раздел 6, така че поправката се потвърждава чрез това, че атаката вече не работи, а не чрез описание.

- 1 Повторно се тества всяка констатация, не извадка. Всичките седем реда в раздел 5 се преразглеждат, включително Medium.
- 2 Тества се поправката, както и обходният път около нея. Проверяваме, че пътят за възпроизвеждане вече се проваля, и търсим очевидното заобикаляне — филтър, приложен на една крайна точка, но не на съседната ѝ, подменени данни за достъп, все още валидни на второ място, права, намалени в една среда, но не в другата.
- 3 Регресия в засегнатия компонент. Там, където една поправка променя удостоверяването, оторизацията или обработката на изхода, повторно задействаме обкръжаващата функционалност, за да потвърдим, че не е отворила нещо ново.
- 4 Ясен резултат за всяка констатация. Всяка констатация се преиздава като затворена, частично отстранена или отворена, с кратка бележка, обясняваща всеки статус, различен от затворен. Нов проблем, въведен от работата по отстраняването, също се докладва, макар повторният тест да не е втора пълна оценка на целия обхват.

Какво заявява писмото за приключване

След повторния тест издаваме подписано писмо за приключване наред с преиздадения отчет. То е написано, за да бъде споделяно: с клиенти, които извършват проверка на доставчици, с одитори, които доказват контрол, и с борд, който иска отговор на една страница. То заявява, без да изисква от читателя да отваря техническия отчет:

Писмото отразява

- Кой е бил тестван, от кого и в кои дати.
- Обхватът и стандартите, които тестването е следвало.
- Колко констатации са били повдигнати и с каква критичност.
- Колко са потвърдени като затворени и датата на повторния тест.
- Всяка констатация, която остава отворена, с нейната критичност и заявения план на клиента.
- Името и подписът на консултанта, който е проверил отстраняването.

Писмото умишлено не отразява

- Стъпки за възпроизвеждане, полезни товари или доказателства.
- Имена на хостове, адреси или архитектурни подробности.
- Всичко, което би помогнало на нападател, който се е сдобил с писмото.
- Каквото и да е твърдение, че средата вече е без уязвимости — то казва какво е било тествано и какво е поправено, и нищо повече.

Защо включваме повторния тест, вместо да го продаваме. Констатация, която е поправена, но никога не е проверена, е план, не резултат — и по нашия опит приблизително разликата между отчет, който променя една среда, и отчет, който бива подреден в папка. Включването на повторния тест премахва търговската причина да бъде прескочена единствената стъпка, която доказва, че работата си е заслужавала.

ПРИЛОЖЕНИЕ А

Използвани инструменти

Инструментите по-долу подпомогнаха ангажимента. Изброени са за прозрачност и за вашите собствени записи; нито един от тях не е произвел констатация сам.

Етап	Обичайни инструменти	За какво беше използвано
Разузнаване и картографиране	Скенери за мрежи и услуги, изброяване през DNS и certificate transparency, уеб обхождащи и откриване на съдържание	Установяване на действителната повърхност за атака: живи хостове, отворени услуги, версии на софтуера, маршрути, параметри и всичко публикувано неволно.
Тестване на уеб и API	Прихващащ прокси с ръчен повторител и fuzzing, конзола за разработчици в браузъра и клиент за повторение на заявки	Ръчно манипулиране на заявки, сравняване на отговори между акаунти и роли и задействане на бизнес логика, която никой скенер не може да моделира.
Оценка на мобилно приложение	Инструменти за извличане на пакети и статичен анализ, скенери за текстови низове и тайни, прихващане на трафик с обработка на сертификати	Разпакетиране на публикувания пакет, възстановяване на вградената конфигурация и тайни и наблюдение какво приложението действително изпраща.
Инфраструктура и Active Directory	Инструменти за изброяване в директорията и за Kerberos, SMB и LDAP клиенти, инструменти за офлайн възстановяване на пароли	Картографиране на домейна, установяване на привилегировани и служебни акаунти и демонстриране на пътища за повишаване на права чрез данни за достъп, офлайн.

Етап	Обичайни инструменти	За какво беше използвано
Проверка и отчитане	Скриптирано повторение на заявки, събиране на доказателства и вътрешен процес за отчитане	Независимо възпроизвеждане на всяка констатация, събиране на доказателства и генериране на този документ и машинно четимия експорт.

За собствени инструменти

Там, където една среда не пасва на готов инструмент — необичаен протокол, специфичен механизъм за удостоверяване, обем данни, който трябва да бъде пресят — пишем каквото е нужно за ангажимента. Всеки скрипт, който докосва средата на клиента, е описан в приложението на действителния отчет и се предоставя при поискване, така че нищо, което е работило срещу вашите системи, да не е непрозрачно за вас.

Инструментите не произвеждат констатации. Всичко в раздел 6 беше проверено на ръка и, където беше безопасно, използвано, за да се демонстрира действително въздействие. Изходът на един инструмент е кандидат за проучване, никога ред в отчета. Кандидати, които се оказаха неизползваеми в контекста, бяха отхвърлени, вместо да бъдат отчетени като констатации.

ПРИЛОЖЕНИЕ Б

Изрично извън обхвата

Какво беше изключено, защо и какво означава това за увереността, която този отчет дава. Всичко изброено тук не беше тествано и нищо в този отчет не бива да се чете като твърдение за него.

Изключено	Причина	Какво означава това за вас
Отказ на услуга и тестване под натиск	Изключено по договореност. Умишленото изтощаване на производствена услуга е неразлично от прекъсване.	Този отчет не казва нищо за това как средата се държи под натоварване или при обемна атака. Тестването на устойчивостта е отделно упражнение.
Физическо проникване	Не е в договорения обхват за този вид ангажимент.	Офиси, центрове за данни, системи за контрол на достъпа с карти и достъп до мрежата на място не бяха оценявани.
Социално инженерство срещу служители на клиента	Изключено по договореност. Фишингът и претекстването се поръчват отделно, за да могат съгласието и работата с данните на служителите да бъдат уговорени самостоятелно.	Човешкият слой не беше тестван. До няколко от констатациите тук нападател, който първо е фишнал, би стигнал по-бързо.
Системи на трети страни и доставчици	Не е наше да го тестваме. Тестването на система, която не притежавате, изисква писменото разрешение на нейния собственик.	Хостваните услуги, доставчиците на плащания и партньорските интеграции бяха третираны като граници. Там, където някоя от тях се появяваше в даден път, пътят беше проследен до границата и спрян.

Исключено	Причина	Какво означава това за вас
Преглед на изходния код	Не беше поръчано за този ангажимент. Тестването беше black и grey box срещу работещи системи.	Дефекти, видими само в изходния код — неактивен път в кода, слаб алгоритъм, който никога не е достиган в тествания поток — не биха излезли наяве тук.
Всичко, което не е названо в подписания обхват	Тестът за проникване е разрешен срещу конкретен списък.	Системи, открити по време на тестването, но извън подписания обхват, бяха докладвани като наблюдения и не бяха изследвани.

Извън обхвата означава нетествано, не сигурно. Това е най-често погрешно разчитаният ред във всеки отчет от тест за проникване. Ако една система не присъства в раздел 2, този документ не твърди абсолютно нищо за нея — нито добро, нито лошо. Там, където едно изключение оставя риск, който ви интересува, разговорът по резултатите е правилното място той да бъде включен в следващата работа.

ПРИЛОЖЕНИЕ В

Как да четете този отчет

Различните читатели имат нужда от различни части на този документ. На никого не му трябва целият, а четенето му от начало до край обикновено е най-бавният начин да се действа по него.

- Ако сте ръководител или член на борда

Прочетете раздел 1 и спрете. Той заявява какво беше тествано, колко сериозен е резултатът и трите неща, които да поправите първо, без жаргон. Раздел 4 обяснява какво означават думите за критичност, ако искате да ги калибрирате.
- Ако отговаряте за програмата по отстраняване

Прочетете раздели 1, 4 и 5, после блока с препоръката на всяка констатация в раздел 6. Заредете машинно четимия експорт във вашата система за проследяване, определете отговорник и дата за всяка констатация и търсете редове, които споделят обща първопричина — тук пет от седем се връщат към данните за достъп и правата.
- Ако сте инженер, който поправя констатация

Отидете направо на вашата констатация в раздел 6. Стъпките за възпроизвеждане ви позволяват да видите проблема, преди да промените каквото и да е, и да потвърдите поправката, като ги изпълните отново и видите, че се провалят. Препоръката е подредена: първата точка е какво да направите днес, останалото е това, което прави поправката трайна.
- Ако сте одитор или клиент, извършващ проверка

Прочетете раздели 2, 3 и 4, за да установите какво беше тествано и как, после раздел 5 за резултата. След отстраняването писмото за приключване в раздел 7 е документът, предназначен за вас: резултатът, без подробностите за атаката.

» Ако решавате дали да поръчате тест

Този документ е предаваният продукт. Това, което бихте получили, е тази структура, запълнена с вашите системи: доказателство за всяка констатация, път за възпроизвеждане, който разработчиците ви могат да следват, поправка, която пасва на вашата технология, среща за представяне на резултатите и повторен тест.

Устройството на една констатация

Всяка констатация в раздел 6 съдържа

- Идентификатор и заглавие — устойчиви, за да може да бъде проследена до приключване.
- Критичност и CVSS v3.1 — оценката по бизнес риск, с резултата и вектора, които са я информирали.
- Засегнат компонент — какво да отидете и погледнете.
- Клас на констатацията — препратка към CWE или техника, за вашата собствена таксономия.
- Описание — какво беше открито и защо има значение.

- Доказателство — записът, който установява констатацията.
- Възпроизвеждане — номерирани стъпки, които вашите инженери могат да следват.
- Въздействие — какво струва на бизнеса, ако бъде използвана.
- Препоръка — конкретна поправка за тази технология, в реда, в който бихме я приложили.
- Статус — отворена при издаването; обновен при повторния тест.

Мярката за един отчет не е колко съдържа. Тя е каква част от него някой може да приложи, без да звъни на автора.

ПРИЛОЖЕНИЕ В · НАКРАТКО

Това е крайният доклад. Сега го поръчайте написан за вашите системи.

Всичко в този примерен документ — доказателствата, пътищата за възпроизвеждане, поправките, повторният тест — е това, което произвежда един ангажимент на PXLSec. Единственото, което липсва, е вашата среда. Кажете ни какво искате да бъде тествано и старши консултант, а не търговец, ще се върне с въпроси, предложен подход и оферта.

- Тестване за проникване — уеб, API, мобилни, облак, вътрешна и външна инфраструктура.
- Оценка на Active Directory и вътрешната мрежа — там, където в действителност започват повечето вътрешни пробиви.
- Red teaming и симулация на противник, отчетени като разказ за това как средата наистина пада.
- Тестване за целите на съответствието по ISO 27001, SOC 2, PCI DSS, DORA и NIS2.
- Повторните тестове са включени по стандарт, защото поправка, която никога не е била проверена, е план, не резултат.

pxlsec.com/bg/contact · hello@pxlsec.com

PXLSecurity LTD. · бул. „Васил Левски“ 12, 1142 София, България · VAT BG203239331 · независими от 2014 г.

Този документ е обобщен примерен документ. Той беше сглобен от няколко анонимизирани ангажимента на PXLSec от различни видове и преписан със синтетични, илюстративни данни. Той не описва нито един действителен клиент. Всички имена използват резервирания домейн example.test и TEST-NET адреси в 192.0.2.0/24; всички акаунти, ключове, записи и оценки по CVSS са измислени за илюстрация. В него никъде не присъства име на клиент, име на хост, адрес, данни за достъп, човек или дата на ангажимент.