



PXLSec · ИЗСЛЕДВАНЕ

Състоянието на нашите констатации 2026

Двадесет и един анонимизирани ангажимента по тестване за проникване и red team, 190 констатации, шест сектора — и какво казват те за това къде наистина стои рискът.

Малък, честен набор от данни, извлечен от предадени отчети на PXLSec. Само числа. Без имена на клиенти, системи, хостове или дати.



КАКВО Е ТОЗИ ОТЧЕТ

Констатациите, без маркетинга

Повечето отчети за „състоянието на сигурността“ съществуват, за да продават продукт. Този е изграден от собствената ни работа по изпълнение. Той обобщава констатациите от 21 скорошни ангажимента на PXLSec — уеб, API, мобилни, вътрешни, външни, red team и фишинг — в шест сектора, и твърди само това, което тези 21 отчета подкрепят.

21

ангажимента

190

констатации

180

оценени по CVSS

6

сектора

Накратко

- 1 Сериозните констатации са редки и решаващи. High и Critical заедно съставляват 31 от 180-те оценени констатации — около 17%. Те са малцинството от обема и на практика целият риск.
- 2 Сериозните изходи не бяха необичайни. Констатация High или Critical се появи в 7 от 21 ангажимента по оценка с CVSS. Отделно, два до три red team ангажимента достигнаха пълно компрометиране на домейна, отчетено като наративен риск.
- 3 Повтарящите се причини са хигиена, не екзотика. Разкриване на информация (76% от ангажиментите), неправилна конфигурация на сигурността (57%) и липсващи заглавки за сигурност (52%) водят в таблицата. Нито една от тях не е изобретателна.
- 4 Чисти резултати се случват и ние ги публикуваме. Едно външно уеб приложение върна нула констатации. Един периметър от около 80 хоста издържа под натиск. Един red team в банка отчете 0% кликове при фишинг.

Метод. Стойностите са взети буквално от обобщението на констатациите на всеки ангажимент. Там, където отчет не е оценявал констатациите по скалата CVSS — някои red team отчети, написани като наративен риск — тези 10 констатации се броят в общия сбор от 190, но са изключени от разпределението по критичност, вместо да бъдат предположени. Всичко е анонимизирано до числа. Извадката е скромна; публикуваме я, защото е действителна, не защото е голяма.

В този отчет

- | | |
|--|--|
| 01 Критичност: тънък слой действителен риск | 02 Къде попаднаха сериозните констатации |
| 03 Единадесетте неща, които продължаваме да намираме | 04 Заглавките: чести, и не те са проблемът |

05 Дефектът, който автоматизацията не може да види

06 Какво тествахме и къде

07 Ангажиментите, при които нищо не поддаде

08 Какво всъщност означава моделът

09 Къде да насочите средствата си

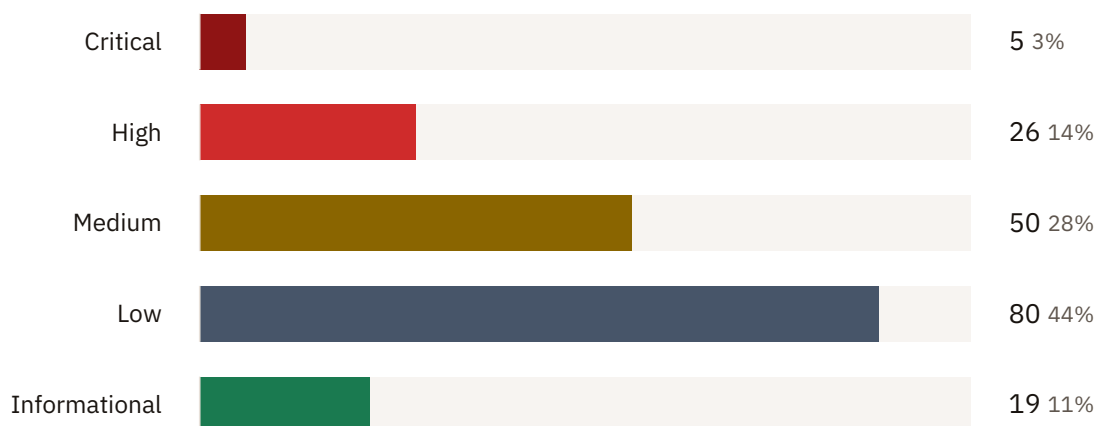
РАЗДЕЛ 01

Критичност: тънък слой действителен риск

От 190 констатации 180 носят оценка, базирана на CVSS. Разпределението е неравномерно и формата му има по-голямо значение от сборовете.



■ Critical 5 ■ High 26 ■ Medium 50 ■ Low 80 ■ Informational 19



Петдесет и пет процента от всичко, което описахме, беше Low или Informational. Това не е запълване — това е структурата на една действителна среда: подробни страници с грешки, остарели заглавки, бисквитка без флаг, банер с версия, който никой не е искал да публикува. Това е и точно диапазонът, до който един скенер може да стигне сам.

31-те констатации High и Critical са друг вид обект. Те почти никога не бяха единични дефекти. Бяха вериги — разкрита тайна, която отключваше служебен акаунт, идентификатор, който можеше да бъде увеличен, хост с два цикъла на обновяване назад, който се оказваше достижим от сегмент, от който не биваше да бъде. Всяко звено поединично беше незабележително. Веригата не беше.

Обемът е в опашката. Рискът е във веригата. Отчет, който само брои констатации, ще ви подведе и за двете.

ЗА ЧЕТЕНЕТО НА ТАБЛИЦА С КРИТИЧНОСТ

Десетте констатации, които не са в това разпределение

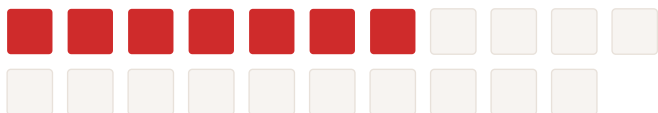
Десет от 190-те констатации идват от red team отчети, написани като наративен риск — маршрут през една среда, а не дефект с вектор. Да бъде наложено число по CVSS върху „достигнахме Domain Admin“ би било измислица, затова тези десет се броят в общия сбор от 190 и са изключени от процентите по-горе. Всичко оценено беше оценено от инженера, който го е открил, по една и съща скала, спрямо въздействието в средата, както е тествана, а не спрямо теоретично най-лош случай. Затова няколко широко известни CVE присъстват в този набор от данни като Low.

РАЗДЕЛ 02

Къде попаднаха сериозните констатации

Броят по критичност ви казва колко. По-полезният въпрос е колко често един ангажимент е произвел нещо, което наистина би провалило една седмица.

7 от 21 ангажимента



Поне една констатация High или Critical, по оценка с CVSS.

2–3 от 21 ангажимента



Red team работа, достигнала пълно компрометиране на домейна, отчетена като наративен риск.

Какво означаваше „сериозно“ на практика

- ❑ Контрол на достъпа, който се доверяваше на идентификатор, подаден от клиента.
- ❑ Данни за достъп и API ключове, оставени там, където приложението може да ги прочете.
- ❑ Служебен акаунт или акаунт към база данни с много повече права, отколкото работата му изискваше.
- ❑ Настройки по подразбиране в Active Directory, които превръщат една опорна точка в целия домейн.
- ❑ Компоненти без поддръжка, все още достижими от място, което имаше значение.

Четете двете мрежи заедно. Само по оценка с CVSS една трета от ангажиментите ни произведоха High или Critical. Добавете red team работата — където предаването е разказ за това как е паднал домейнът, а не таблица с оценени дефекти — и наистина сериозен изход се оказва по-близо до половината от всички ангажименти тук. Нито един от тях не изискваше новаторско използване на уязвимост; нито един не зависеше от zero-day. Те зависеха от оторизацията, от тайните и от хигиената на директорията.

Active Directory attack path — service ticket request

SYNTHETIC EXAMPLE

A kerberoastable service account turns a low-privilege foothold into database administrator.

```
operator@jump-01 ~$ bash

operator@jump-01:~/ops$ GetUserSPNs.py example.test/alice -dc-ip 192.0.2.10 -request
Impacket — synthetic transcript, invented values throughout

[*] Enumerating accounts with servicePrincipalName set
ServicePrincipalName      Name      MemberOf
-----
MSSQLSvc/db-01.example.test  svc_sql   CN=Service Accounts
HTTP/app-01.example.test    svc_web   CN=Service Accounts

[*] Requesting TGS for svc_sql — ticket encrypted with the account password
$krb5tgs$23$*svc_sql$EXAMPLE.TEST$MSSQLSvc/db-01.example.test*$
3f9a1c0b7e2d4a8f...<truncated — fabricated value>

operator@jump-01:~/ops$ hashcat -m 13100 tgs.txt wordlist.txt
Recovered.....: 1/2  svc_sql:<example-password>

operator@jump-01:~/ops$
```

Attack path

low-privilege user → kerberoastable SPN → offline password crack → db-01 administrator

i Illustrative example — synthetic data, not client evidence.

Илюстративна реконструкция на формата, която приема вътрешното повишаване на права: служебен акаунт, подлежащ на roasting, разбиване офлайн, повторна употреба срещу привилегирована група. Синтетични данни; не е изобразена клиентска среда.

РАЗДЕЛ 03

Единадесетте неща, които продължаваме да намираме

Броенето на констатации възнаграждава шумните среди. Броенето на ангажименти, в които даден клас се е появил поне веднъж, е по-трудно за заиграване — затова това измерва тази таблица, във всички 21 ангажимента.



Лентите са оцветени според типичната критичност, която класът е носил в този набор от данни: ■ предимно Low и Medium ■ често High ■ откъдето дойдоха нашите Critical. Процентите са дялът от 21-те ангажимента, затова не се сумират до 100.

Началото на тази таблица е почти скучно. Разкриването на информация се появи в три ангажимента от четири: стекови следи, индекси на директории, версии на софтуер, вътрешни пътища в коментари. Неправилната конфигурация и липсващите заглавки следват плътно. Това са настройки по подразбиране, които никой не е избирал и за които никой не отговаря.

Долната част на таблицата е там, където е вредата. Разкритите тайни, повишаването на права и слабостите в Active Directory се появиха в по-малко ангажименти — по четири до шест всяка — но произведоха непропорционален дял от констатациите High и Critical. Честотата и въздействието вървят в противоположни посоки, което е точно причината подредена по брой опашка за отстраняване да е погрешната опашка.

Подредете списъка си със задачи според това колко често се появява нещо и ще поправите първо заглавките. Подредете го според това колко ви струва, когато бъде използвано, и ще поправите първо тайните.

ЗА ПРИОРИТИЗИРАНЕТО НА СПИСЪК С КОНСТАТАЦИИ

РАЗДЕЛ 04

Заглавките: чести, и не те са проблемът

Липсващи заглавки за сигурност се появиха в 11 от 21 ангажимента, което ги поставя на трето място в таблицата. Почти всеки път бяха оценени като Low или Informational, което ги поставя близо до дъното по риск. И двата факта са верни едновременно, а разликата между тях е мястото, където немалка част от бюджета за сигурност отива да умре.

Причината заглавките да доминират таблиците с честоти е, че са тривиално видими. Една заявка разкрива целия набор, без акаунт, без договаряне на обхват и без нужда от преценка. Тази видимост ги прави съблазнителен заместител за зрялост по сигурността, особено за всеки, който няма да чете след първата страница на отчета. Това е лош заместител. Безупречен набор от заглавки стои напълно спокойно върху модел на оторизация, който се доверява на каквото изпрати клиентът, и ние сме тествали точно такава подредба повече от един път.

По-лошо, наличието не е действие. Една заглавка може да бъде зададена, връщана при всеки отговор, и да не прави абсолютно нищо — опровергана от втора политика, насочена към погрешен произход, или просто игнорирана от браузърите във вида, в който е написана. Скенер, който проверява за името на заглавката, отчита успех. Контролът все още отсъства.

A control that is present but inert
Two framing policies ship together. Only one of them is applied.

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Content-Type: text/html; charset=utf-8
4 Strict-Transport-Security: max-age=31536000
5 X-Content-Type-Options: nosniff
6 X-Frame-Options: SAMEORIGIN
7 Content-Security-Policy: default-src 'self';
8   frame-ancestors *
9 Referrer-Policy: strict-origin-when-cross-origin
10 Cache-Control: no-store
```

IGNORED BY THE BROWSER
X-Frame-Options is discarded whenever the response also carries a frame-ancestors directive.

HONOURED BY THE BROWSER
frame-ancestors * permits any origin to frame the page, so the stricter SAMEORIGIN header has no effect at all.

Net effect: the page can be framed by any site. A header scan reports framing protection as present.

Illustrative example — synthetic data, not client evidence.

Илюстративно: правилните имена на заглавки, налични и бездействия. Сканиране, което търси имена, а не действие, отчита този отговор като закален. Синтетични данни; не е изобразена клиентска система.

Затова ги задайте. Евтини са, бързи са и наистина намаляват обхвата на пораженията от дефект, който съществува другаде. Само не четете чист отчет за заглавките като доказателство, че дефектът го няма — и не позволявайте един ред лесни задачи да измести трудните.

Заглавката е предпазен колан. Заслужава си да се носи, евтина е и нито веднъж не е спряла никого да се забие в стена.

РАЗДЕЛ 04 · НАКРАТКО

Как се тества както трябва. Питайте какво прави браузърът със заглавката, а не дали заглавката я има. Противоречиви политики, погрешен обхват и остарял синтаксис — всички връщат зелена отметка при проверка по име.

РАЗДЕЛ 05

Дефектът, който автоматизацията не може да види

Нарушен контрол на достъпа се появи в 8 от 21 ангажимента — около половината от честотата на разкриването на информация. Въпреки това той остана най-надеждният източник на констатации High в работата по приложения в целия този набор от данни.

Моделът се повтаря почти без изменение. Сървърът удостоверява извикващия правилно — сесията е истинска, токенът е валиден, входът е бил действителен — и след това взема идентификатора на обекта направо от заявката и му се доверява. Сменяте числото, прочитате чужд запис. Интерфейсът никога не е предлагал този запис, така че от гледна точка на приложението заявката не би трябвало да съществува. Тя все пак съществува.

Това е невидимо за автоматизацията по структурна, а не по техническа причина. Заявката е правилно формирана. Сесията е валидна. Отговорът е законен 200 с правилно оформено тяло. Нищо в обмена не е необичайно, защото нищо в обмена не е грешно — освен кой има право да го направи. Да бъде уловено това означава да се знае на кои записи акаунтът има право и да се забележи, че му е бил подаден запис, на който няма право — което на практика означава тестер с два акаунта, който внимава.

Broken object-level authorisation
The session is valid, so the object id is simply trusted.

SYNTHETIC EXAMPLE

REQUEST · authenticated as alice

```
1 GET /api/v1/invoices/10423 HTTP/1.1
2 Host: app.example.test
3 Authorization: Bearer eyJhbGciOi...<truncated>
4 Cookie: session=sess_a1b2...<truncated>
5 Accept: application/json
```

→ object id taken straight from the path

→ session belongs to alice (tenant A)

RESPONSE

```
1 HTTP/1.1 200 OK
2 Content-Type: application/json
3
4 {
5   "invoice_id": 10423,
6   "tenant": "tenant-b",
7   "owner": "bob@example.test",
8   "amount_due": "4150.00"
9 }
```

→ request succeeds

→ record belongs to a different tenant

→ not the caller

What the figure shows
The authorisation layer confirms that a session exists, not that the session owns invoice 10423. Incrementing the identifier walks the whole billing table across tenants.

Illustrative example — synthetic data, not client evidence.

Илюстративно: една удостоверена сесия, един увеличен идентификатор и 200 там, където мястото беше на 403. Синтетични данни; не е изобразена клиентска система.

Поправката е структурна, не крѣпка

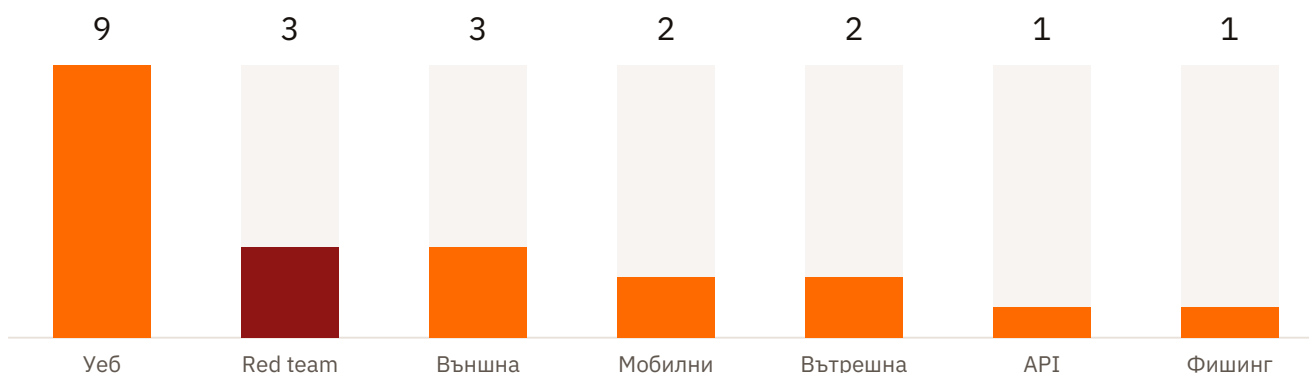
Всяка една от тези констатации беше затворена по един и същ начин: преместване на проверката на сървъра, оценяването ѝ спрямо сесията, а не спрямо заявката, и прилагането ѝ към всеки отделен обект и действие, а не към маршрута. Скриването на бутона не е контрол. Нито пък труден за отгатване идентификатор.

РАЗДЕЛ 06

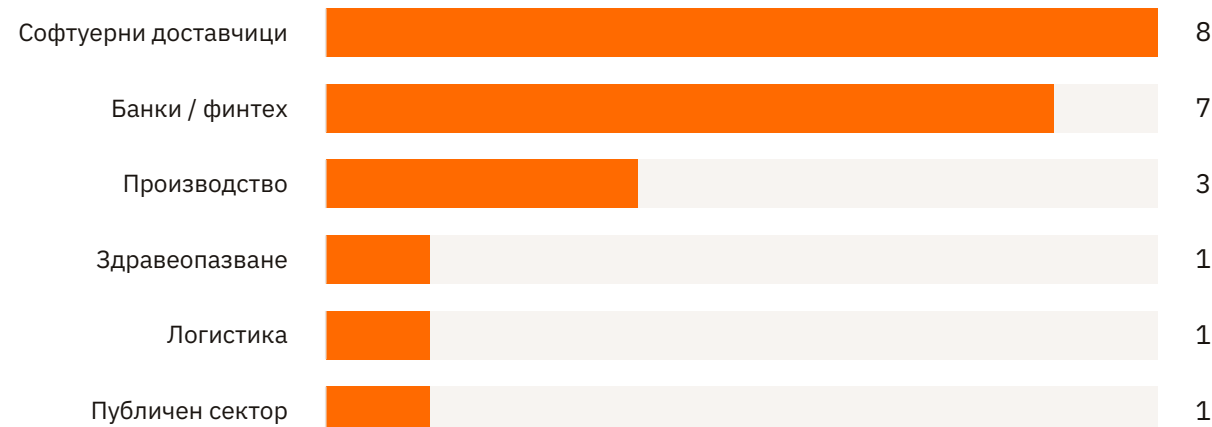
Какво тествахме и къде

Набор от данни с този размер се определя от състава си, затова ето го в пълния му вид. Работата по уеб приложения доминира; red team ангажиментите са малко, но носят по-голямата част от наративния риск, описан по-горе.

По вид ангажимент



По сектор



Две уговорки следват директно от този състав. Първо, разпределението по критичност е наклонено към тестване на уеб и външни приложения, защото това бяха повечето от тези ангажименти; набор от данни с превес на вътрешното тестване би изглеждал по-суров. Второ, софтуерните доставчици и финансовите услуги заедно съставляват 15 от 21-те ангажимента, така че констатациите за Active Directory и Kerberos идват от тесен срез на извадката и трябва да се четат като показателни, а не представителни.

Един ангажимент означава

Един подписан обхват, предаден като един отчет, независимо колко хоста, приложения или седмици е включвал. Повторните тестове не се броят отново.

Умишлено отсъстващо

Без имена на клиенти, хостове, адреси, лични данни или дати. Обозначенията на секторите са умишлено широки по същата причина.

РАЗДЕЛ 07

Ангажиментите, при които нищо не поддаде

Тест, който не открива нищо, не е напразен тест и не е провален тест. Три резултата в този набор от данни си заслужава да бъдат изречени направо, защото именно те са онези, които отчет, писан от доставчик, тихо би пропуснал.

✓ Нула констатации

Едно външно уеб приложение не произведе никакви констатации. CDN и WAF пред него се държаха според документацията, CAPTCHA издържа под автоматизация, а ограничаването на честотата се прилагаше там, където трябваше. Приложихме цялата методология за приложения и го описахме като чисто.

✓ Без използваеми High или Critical

Логистичен периметър от около 80 хоста не върна нищо използваемо на ниво High или Critical. Седем широко известни CVE, отбелязани чрез откриване на версия, бяха доказани като неизползваеми в контекста — което само по себе си е констатация, и по-полезна от седемте задачи, които иначе биха били създадени.

✓ 0% кликове при фишинг

При един red team в банка нито един получател не кликна. Втора кампания никога не достигна пощенска кутия — беше спряна на шлюза. Когато все пак установихме опорна точка по друг начин, мрежовата сегментация издържа и хоризонталното придвижване спря там, където проектът предвиждаше.

Проверен чист резултат е доказателство. Непроверен е надежда. Разликата е в това дали някой действително е атакувал въпросното нещо.

ЗА ОТЧИТАНЕТО НА ДОБРИ НОВИНИ

Отчитаме тези резултати толкова внимателно, колкото отчитаме и Critical констатациите, по практична причина: контрол, който е бил атакуван и е издържал, е единственият вид, около който можете да планирате. Доказването, че седем CVE не се прилагат, спестява повече инженерно време от намирането на осми, който също не се прилага.

Тук има търговско изкушение, което си заслужава да бъде назовано. Отчет без констатации изглежда, за всеки, който го прегледа бързо, като отчет, за който не си е струвало да се плаща. Стимулът е той да бъде запълнен — флагът на бисквитката и липсващата заглавка да бъдат вдигнати до Medium и ангажиментът да бъде наречен продуктивен. Ние ги броим като това, което са — затова 55% от този набор от данни се намира в двата най-ниски диапазона по критичност и затова ангажиментът с нула констатации е отчетен като нула.

Какво все пак ви дава чистият резултат. Тестван и непробит контрол е единственият контрол, около който можете да планирате. Той свива следващия обхват, отменя допускания, които тихо са носели риск, и превръща списък от теоретични CVE в списък от такива, които действително се отнасят за вас.

РАЗДЕЛ 08

Какво всъщност означава моделът

При 21 ангажимента и 190 констатации три заключения издържат въпреки малката извадка. Те не са изненадващи. Точно това е смисълът — проблемите, които се повтарят, са онези, които всички вече познават.

1 · Дефектът, който ви навреди, сам по себе си беше скучен

Всяка констатация High и Critical в този набор от данни беше сглобена от части, всяка от които поединично би била оценена като Low. Банер с версия е Informational, докато не каже на нападателя кой необновен билд да атакува. Подробна грешка е Informational, докато не изтече вътрешното име на хост, което трябва на следващата заявка. Критичността е свойство на веригата, а веригите се намират от хора, не от скенери.

2 · Хигиената е проблем на капацитета, не на знанието

Никой, когото тествахме, не беше в неведение, че компонентите трябва да се обновяват или че заглавките трябва да се задават. Повтарящите се класове в началото на таблицата ни с честоти са онези без ясен отговорник: отклонение в конфигурацията, зависимост, наследена от шаблон, хост, принадлежащ на екип, който вече не съществува. Повече указания не поправят тези неща. Отговорен инвентар — да.

3 · Контрол върху диаграма не е контрол в трафика

Ангажиментите, които издържаха, не притежаваха по-малко уязвимости в някакъв абстрактен смисъл. Те имаха работещи контроли по пътя на атаката: WAF, който наистина е в линията, ограничаване на честотата, което наистина се налага, сегментация, която наистина прекъсва маршрута. Тестването е начинът да разберете кой от двата вида притежавате.

HOW A FINDING APPEARS IN A PXLSec REPORT

SAMPLE FINDING — NOT A REAL ENGAGEMENT

HIGH CVSS 8.1 PXL-2026-014

Invoice records readable across tenant boundary

Insecure direct object reference in the billing API

AFFECTED COMPONENT	HOST	TESTED FROM	STATUS
GET /api/v1/invoices/{id}	app.example.test	192.0.2.44 (TEST-NET-1)	Open — retest pending

IMPACT

Any authenticated tenant can read every other tenant's invoices by changing one number in the URL. The exposed fields include billing contacts, amounts outstanding and payment status.

FIX

Resolve the object from the caller's own tenant scope and return 404 when the identifier falls outside it. Add a regression test that asserts a cross-tenant identifier is never served.

Evidence: synthetic request/response pair

PXLSec · State of Our Findings 2026

Illustrative example — synthetic data, not client evidence.

Илюстративно: структурата, по която беше написана всяка констатация в тези ангажименти — доказателство, въздействие, възпроизвеждане, поправка, повторен тест. Синтетично съдържание.

РАЗДЕЛ 09

Къде да насочите средствата си

Ако този набор от данни е единственото доказателство, с което разполагате, той подкрепя определен ред на работа. Той е подреден според въздействието, което действително наблюдавахме, а не според това колко често се е появявал класът.

- 1 Тайните и правата, първо. Разкритите тайни и повишаването на права се появиха съответно само в 6 и 5 ангажимента, а помежду си произведоха непропорционален дял от най-лошите ни констатации. Извадете данните за достъп от изходния код, конфигурацията и пакетите от страна на клиента, след което свалете всеки служебен и всеки акаунт към база данни до правата, които работата му изисква. Това е най-високото съотношение между отстранен риск и вложено усилие в целия този отчет.
- 2 Оторизация от страна на сървъра за всеки обект и всяко действие. Нарушеният контрол на достъпа беше най-надеждният източник на констатации High в работата по приложения. Удостоверяването не е оторизация. Проверката принадлежи на сървъра, спрямо сесията, за всеки отделен обект — а не в интерфейса, който скрива бутона.
- 3 Хигиена на Active Directory. Подписване и обвързване на канала, акаунти, изложени на Kerberoasting и AS-REP, и прочистване на привилегированите групи. Четири ангажимента извадиха наяве проблеми в този клас и те бяха решаващи в red team работата. Това са най-евтините вътрешни победи с голямо въздействие и повечето от тях са въпрос на конфигурация, а не на покупка.
- 4 Обновяване и компоненти без поддръжка. Неподдържаният софтуер стои зад голям дял от констатациите Critical в този набор от данни. Трудното никога не е крпката; трудното е да се знае, че хостът съществува, и кой отговаря за него. Инвентар преди ритъм.
- 5 Жизнен цикъл на токените. Краткотрайни, отменени от сървъра, с ограничен обхват и никога в URL, в журнал или в скрипт на страницата. Управлението на сесии и токени се появи в 6 ангажимента и провалите бяха последователно в жизнения цикъл, а не в криптографията.





PXLSec · ЗАЯВЕТЕ ПРЕДЛОЖЕНИЕ

Всичко в този отчет произлиза от ръчно тестване от старши инженери — същата работа, която бихме приложили към вашите системи. Доказателство за всяка констатация, път за възпроизвеждане, който разработчик може да следва, поправка, която пасва на вашата технология, и повторен тест, който доказва, че е проработила.

-
- Тестване за проникване — веб, API, мобилни, вътрешна и външна инфраструктура.
 - Red teaming и симулация на противник, отчетени като разказ за това как средата действително пада.
 - Сигурна разработка и преглед на кода, преди констатацията да съществува.
 - Повторните тестове са включени, защото поправка, която никога не е била проверена, е план, не резултат.

pxlsec.com/bg/contact ·

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

Този документ не съдържа имена на клиенти, хостове, адреси, лични данни или дати на ангажименти. Всички числа са обобщени стойности, взети буквално от предадени отчети. Илюстративните фигури са синтетични и не изобразяват действителна среда.