CASE STUDY

JSCRAMBLER HELPS NEOBANKS PROTECT JAVASCRIPT

SKIP AHEAD IN THE BANKING RACE

"Protecting our JavaScript was a requirement from day one. Investors and management made sure that it was a priority. Today, not a single product ships without secure client-side logic and this has been extremely effective."

jscrambler

NEOBANKS
FROM INNOVATION TO SATISFACTION

Neobanks defy traditional banking by betting everything on digital and delivering customer-centric services for payments and money management. Today, 73% of all **consumer interactions with banks are done digitally**. While traditional banks have invested in Web and mobile platforms, Neobanks release twice as many new features and three times more app updates per year. They also run 42% faster than incumbents. As a result, user satisfaction ratings for Neobanks in the US (90%) are much higher than that of traditional banks (66%).

Neobanks' technological flexibility comes from investing in cloud-based infrastructure and advanced web and mobile applications using **modern JavaScript frameworks such as React Native**. With this approach, they cut product development cost and time, paving the way for rapid iteration and innovation. This is greatly aided by relying on third-party integrations instead of having to develop every piece of code in-house.

FASTER DEVELOPMENT, LARGER ATTACK SURFACE

In software development, pursuing agility and speed often means **widening security gaps**. Despite JavaScript's numerous advantages, Neobanks must be aware that client-side JavaScript is exposed and can originate attacks — including **intellectual property theft, code tampering, application abuse, and data exfiltration**. Unless protected with an enterprise-grade solution, this exposed JavaScript poses a key business threat.

SUMMARY

CHALLENGES

Launch secure web and mobile apps to market

Conceal and actively protect client-side logic

Prevent application tampering

Keep user data safe

Comply with relevant regulations

SOLUTION

Enterprise JavaScript Protection:

Polymorphic obfuscation to conceal client-side logic

Self-Defending to mitigate debugging and tampering

Robust countermeasures to automatically stop attackers

RESULTS

Seamless app build integration

Met OWASP recommendations

Increased PSD2 compliance

New competitive advantage for funding rounds

0 issues with QA tests

0 successful attacks to JavaScript

CHALLENGES

Over the last 12 months, Neobanks from North and South America, Europe, and Asia have come to Jscrambler with significant security challenges. With Web and mobile apps built using JavaScript — and a strong incidence of cross-platform frameworks for mobile development like React Native and Ionic — security teams understood early-on that client-side logic would represent a significant security liability.

There was a high likelihood of having to run sensitive logic on the client-side, and so it became paramount to guarantee that this logic would be concealed with the most potent and resilient technology available today.

It was also mandatory to ensure that automated reverse-engineering tools would always fail to reverse the concealed code, while making it extremely unfeasible for attackers to achieve it manually.

As these Neobanks' apps would handle valuable services, another key challenge was guaranteeing that malicious actors wouldn't be able to tamper with the code. **JavaScript had to react in runtime to mitigate these attacks.**

And since both the Web and mobile apps would be handling sensitive data — credentials, personally identifiable information, financial details — an additional pre-eminent requirement was guaranteeing that JavaScript couldn't serve as a gateway for attackers to layout attacks that would steal user data.

> *"We're called "challenger banks" for a reason. One of our toughest challenges is still gaining customer trust. When handling their data, we can't just meet the minimum requirements — **we must excel at it and keep user data safe at all costs**."*

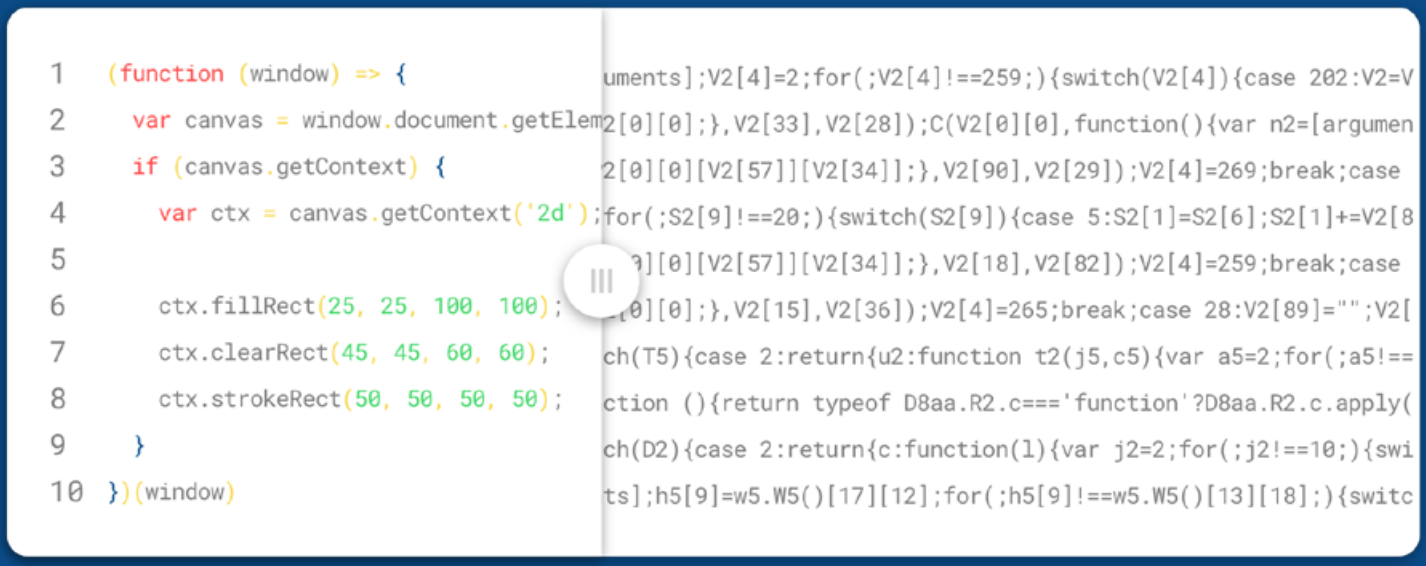
With each Neobank possessing more than one application, it was also essential to guarantee that JavaScript protection would fit seamlessly into their CI/CD and integration tests.

Finally, in such a heavily regulated sector, another significant challenge was achieving compliance with regulations such as PSD2 with a special focus on client-side attacks.

SOLUTION

To meet the highest standards for JavaScript protection, these Neobanks sought a holistic on-premise solution that would fit their processes and scale. Here, Jscrambler presented a mature and proven client-side security product suite that — much as Neobanks themselves — is defined by continuous innovation.

The first step towards securing JavaScript was Jscrambler's polymorphic obfuscation. With this critical security layer, all of the source code of Neobanks' apps was concealed beyond possible recognition. Here, Jscrambler's set of the most potent and resilient transformations was key to guarantee cutting-edge obfuscation. Its inherent polymorphism ensured that each new code deploy would be completely different — an extra line of defense against reverse engineering attempts.



```
(function (window) {
  var canvas = window.document.getElem
  if (canvas.getContext) {
    var ctx = canvas.getContext('2d');

    ctx.fillRect(25, 25, 100, 100);
    ctx.clearRect(45, 45, 60, 60);
    ctx.strokeRect(50, 50, 50, 50);
  }
})(window)
```

```
uments];V2[4]=2;for(;V2[4]!==259;){switch(V2[4]){case 202:V2=V
2[0][0];},V2[33],V2[28]);C(V2[0][0],function(){var n2=[argumen
2[0][0][V2[57]][V2[34]];},V2[90],V2[29]);V2[4]=269;break;case
for(;S2[9]!==20;){switch(S2[9]){case 5:S2[1]=S2[6];S2[1]+=V2[8
0][0][V2[57]][V2[34]];},V2[18],V2[82]);V2[4]=259;break;case
[0][0];},V2[15],V2[36]);V2[4]=265;break;case 28:V2[89]="";V2[
ch(T5){case 2:return{u2:function t2(j5,c5){var a5=2;for(;a5!==
ction (){return typeof D8aa.R2.c==='function'?D8aa.R2.c.apply(
ch(D2){case 2:return{c:function(l){var j2=2;for(;j2!==10;){swi
ts];h5[9]=w5.W5()[17][12];for(;h5[9]!==w5.W5()[13][18];){switc
```

Original

Obfuscated

"The concealed code looks like absolute nonsense and passed all of our tests. Being able to pick from dozens of well-documented transformations and fine-tune each one was very important."

Following obfuscation, these Neobanks leveraged an additional Jscrambler security layer to meet the challenges of preventing application tampering and client-side data exfiltration: self-defending. With this runtime protection, their apps gained a series of integrity checks that detect every debugging attempt and also break the app whenever tampering occurs. Taking advantage of other client-side countermeasures, such as calling a custom function, has enabled these Neobanks to further stop malicious users.

Neobanks' Security Engineers were well aware of the problem and the required steps for solving it. After the initial setup of their Jscrambler instance, it took on average 2 weeks and less than 3 meetings with Jscrambler's Engineers to integrate Jscrambler seamlessly into their CI/CD pipeline. From there, Jscrambler became an automated part of their application build process.

“The Jscrambler team has extensive knowledge of JavaScript. Communication with our engineering teams was excellent and all issues were solved quickly.”

RESULTS

5 WEB AND MOBILE APPLICATIONS SECURED IN 3 WEEKS

Securing JavaScript code, first and foremost, requires awareness of the threats caused by having important logic exposed on the client-side. Neobanks clearly have this pain from the very onset of the business, as their main assets depend upon it.

By opting for Jscrambler’s proven JavaScript protection technology, product teams met their main requirement of integrating a code protection solution seamlessly into their CI/CD. Now, these Neobanks deploy secure code to production knowing that each build has a fresh set of the most potent and resilient JavaScript protection available today.

"Today, not a single product ships without secure client-side logic and this has been extremely effective."

In parallel, security teams were able to fulfill security recommendations by OWASP, namely the OWASP Mobile Top 10, which states "in order to prevent effective reverse engineering, you must use an obfuscation tool" and "The app must be able to react appropriately at runtime to a code integrity violation". They also became compliant with several PSD2 mandates, namely regarding transaction monitoring and strong customer authentication.

To management, ensuring that their applications' source code was protected against reverse engineering and tampering ultimately meant a new competitive advantage. Keen investors are aware of the liability posed by exposed JavaScript in Neobanking; with Jscrambler, Neobanks gained the upper hand in future funding rounds and the trust of millions of potential customers.

In an industry where numbers are everything, for these Neobanks, the outcome of integrating Jscrambler couldn't be rounder:

0 INTEGRATION ISSUES
0 SUCCESSFUL ATTACKS TO JAVASCRIPT CODE

jscrambler

CONTACT US

If you want to know more about how Jscrambler can help you Secure your JavaScript and HTML5, don't hesitate to contact us

hello@jscrambler.com

+1 **650 999 0010**

jscrambler.com

Gartner®

Jscrambler is the leader in Client-Side Application Security

Recognized in Gartner's **Market Guide for Online Fraud Detection**

and in Gartner's **Market Guide for In-App Protection**

Trusted by the Fortune 500 and 43000+ companies and individuals globally.