Introduction

This document aggregates responses to the TAG's Self-Review Questionnaire: Security and Privacy for the current versions of these Web Payments Working Group specifications: Web Payments API, Basic Card Payment spec, and Payment Method Identifiers. The authors of this report have aggregated responses rather than analyzing each document in isolation to catch any relevant interaction among them.

This report was prepared for the discussion at the Working Group's July 2016 face-to-face meeting. Adam Roach, Evgeny Vinogradov, and Ian Jacobs contributed to this report.

Analysis

The following sections correspond to sections in the Self-Review Questionnaire: Security and Privacy document.

3.1 Does this specification deal with personally-identifiable information?

Yes, and quite a bit of it. The payment request API allows a merchant site to request a shipping address (including both personal and business names) as well as payer email and phone number fields. The Basic Card Payment specification collects card holder names, credit card numbers, and billing addresses.

These data are all user-entered and, based on expected browser implementation, will require explicit user consent to be sent to a web site. This consent will likely come in either the form of a per-transaction explicit interaction, or in the form of a persistent, user-set preference to always send such information automatically to specific named origins.

Recommendation

The "Privacy Considerations" section of the Payment Request API spec includes a one-sentence treatment of this issue. This should be expanded to address the types of consent (contemporaneous consent versus stored consent), and make it clear that nearly all data exposed through the API is personally identifying.

3.2 Does this specification deal with high-value data?

Being used to perform financial transactions, the Payment API specification and Basic Card Payment specifications inherently deal with high-value data.

Currently, the Basic Card Payment specification has both text and diagrams that strictly encourage web sites to store credit card information for later use. While this is common practice for web sites today, it is generally done to reduce friction of the users making subsequent purchases from the same merchant. This is done as a calculated trade-off: the increase in sales due to lower checkout friction is considered to be of greater value than the liability exposure of retaining possession of a large store of credit card information.

In the age of increasing corporate data breaches, maintaining a database of this kind of high-value data on a network-attached server is becoming increasingly imprudent. With the introduction of the Payment Request API, however, websites are being given an opportunity to request the information from the browser in an automated fashion every time it is needed for a transaction, and with no more friction than if it were stored by the merchant site.[1]

Recommendation

We suggest that the Basic Card Payment specification strongly discourage web sites from storing credit card information for future use, except in the case of future or recurring payments. We suggest including explicit guidance that in such cases, Web site owners should take careful action to prevent disclosure. The sequence diagram in the document should similarly be updated so as not to encourage server-side credit card information storage.

Because of the potential for such storage by web sites, and because of the potential for web browser state synchronization (usually assisted by a synchronization server), we also tentatively recommend that the Basic Card Payment specification make some level of mention of PCI DSS compliance. We propose that the group seek input from major credit card processing companies -- such as Visa and American Express -- regarding what language about PCI DSS compliance (if any) is appropriate for the specification. Here is the sort of statement we have in mind: “The privacy and security sections in this document do not replace conformance with PCI DSS or any other regulations. Implementors and users of the payment APIs

[1] This also provides a couple of non-security-related benefits to users. In particular: users are no longer required to update their credit card numbers and expiration dates at a variety of websites when issued a new card, and the involvement of a configurable user agent allows the user whatever degree of control they wish to have over the sending payment data to websites.

should determine whether they are also subject to PCI DSS and/or other legal regulations.”

If we elect not to discuss PCI DSS compliance in the Basic Card Payment specification, we strongly recommend mentioning the topic by name and explicitly indicating that it will not be discussed.

3.3 Does this specification introduce new state for an origin that persists across browsing sessions?

The reviewed documents do not explicitly introduce any long-lived origin-scoped state. Payment apps may choose to make use of existing web technologies for state persistence.

If users allow specific sites to access persistent payment information without contemporaneous user consent, those permissions will presumably be stored on a per-origin basis. The sites themselves will have no programmatic way of setting such information; however, if such permissions *are* set, sites can easily detect such permission by observing the length of time between a payment request and a payment response.

Recommendation

None.

3.4 Does this specification expose persistent, cross-origin state to the web?

In general, the information that can be passed to and from payment applications is specified on a per-payment-method basis. Since the payment app controls the information it presents to the users, any un-policed use of this information can trivially result in the payment provider acting as a cross-origin correlator. Without mediator policing of values passed between merchant sites and payment providers, this can happen intentionally through collusion with the payment provider. Even with the specification and enforcement of allowed fields, such correlation may occur unintentionally due to overly-permissive fields defined for the use of payment apps.

Further, for the general case, mediator-provided information -- such as phone number and email address -- provide strong, unchanging, mostly-unique correlators for people that are invariant across origins.

The use of payment app registration may provide an additional means of uniquely identifying browser instances. Because a payment request for an unregistered payment type ID will fail immediately, while a request for a registered one will wait for user interaction, websites can use promise resolution timing to probe whether a specific payment type ID is installed in a browser. Because the API provides a means to cancel an outstanding payment request, attackers can wait until the elapsed time has passed a chosen confidence interval and then cancel the request, eliminating the need for user interaction. Finally, since payment apps can register for multiple payment method identifiers, payment apps can register for a unique combination of these identifiers, which web sites can then use this technique to probe.

For example, if a payment app registered for some random subset of "https://example.com/0" through "https://example.com/7", then websites would be able to add eight bits of fingerprinting to any other techniques they may currently employ. Even stronger, if the site were looking for a specific user, registering a unique payment type ID (e.g. http://example.org/02332fb8-24b8-44bd-91be-cef3ef849926) allows use as limited but guaranteed-unique supercookie.

Recommendation

The Basic Card Payment specification provides additional information that forms a unique correlator. Unlike unique, cross-origin information provided by the Payment Request API, the `cardNumber` field provided by the Basic Card API is (necessarily) required rather than optional. The current document appears to assume that any Basic Card Payment app will request all possible fields; however, there is a PR filed for leaving off unneeded information. For the sake of privacy, we recommend this PR be accepted.

The foregoing facts put additional emphasis on the importance for explicit user consent in providing payment or any information associated with payments to a requesting web site. We suggest including the following guidance: any mechanism that allows users to persistently grant information should take steps to inform users that doing so will allow various websites to positively identify and correlate a user, even across site owners.

3.5 Does this specification expose any other data to an origin that it doesn't currently have access to?

Data is shared between origins (payment app and merchant site) using this API, but always with user consent. Beyond those issues already discussed in previous

sections, we do not believe these specifications introduce any additional attack surface other than that already possible.

Recommendation

None.

3.6 Does this specification enable new script execution/loading mechanisms?

The overall system design includes the concept of payment app registration. While the design for this is currently somewhat fluid, it will likely involve loading JS code in a sandboxed context with communication between that payment app and merchant web sites. This JS code will (presumably) run in the origin from which it was loaded, preserving same-origin protections.

Recommendation

Revisit this question after the Working Group invocation approaches in the Payment App API specification.

3.7 Does this specification allow an origin access to a user's location?

The Payment Request API provides for a means of collecting mailing address, which is frequently correlated with a user's residence and therefore location. The ability to collect phone numbers provides a similar, if looser, capability, as components of phone numbers typically correspond to countries, cities, and individual exchanges (although the advent of number portability and mobile devices dilutes this somewhat).

The Basic Card specification allows for the collection of billing addresses, which have mostly the same properties as shipping addresses. It also provides credit card numbers; the initial six digits of credit card numbers can be used to identify the country of the issuing bank, which will typically correlate to the user's home country.

Recommendation

None.

3.8 Does this specification allow an origin access to sensors on a user's device?

No. Individual native payment applications may access such devices (e.g., using mobile phone cameras to read credit card numbers, magnetic stripe/chip readers to read cards, or biometrics to authenticate users), but those behaviors are outside the purview of the reviewed specifications.

Recommendation

None.

3.9 Does this specification allow an origin access to aspects of a user's local computing environment?

The Payment Request API, in conjunction with the anticipated Payment App API, allows access to installed payment apps -- both browser-based and native. This allows for programmatic determination of whether certain apps are installed (e.g., if the Bobpay mobile app registers with installed web browsers, then web pages can probe to determine whether users have the Bobpay mobile app installed -- see the response to section 3.4 for a discussion how how this probing can take place).

Recommendation

None.

3.10 Does this specification allow an origin access to other devices?

No. See also the response to section 3.8, as it is possible to imagine scenarios such as bluetooth-attached devices.

Recommendation

None.

3.11 Does this specification allow an origin some measure of control over a user agent's native UI?

Through the Payment Request API, web pages indicate supported payment methods and instruct the browser to collect additional information from the user. These options will impact the native dialog (or other UI construct) used by the browser to allow users to select a preferred payment app and to enter/approve the requested additional information. These UI changes are expected to be scoped exclusively to the payment flow UI, and should not otherwise affect the browser's appearance.

Recommendation

None.

3.12 Does this specification expose temporary identifiers to the web?

No. These specifications expose identifiers to the web (as described in previous sections), but those identifiers are not temporary.

Recommendation

None.

3.13 Does this specification distinguish between behavior in first-party and third-party contexts?

The Payment Request API has an ongoing discussion regarding whether the API can be invoked in a third-party context. See the github issue "Should the Payment Request API only be available in a top-level browsing context?" and the minuted discussion from the 12 May 2016 interim meeting.

Recommendation

Revisit this question after the group addresses the security issue about top-level browsing context.

3.14 How should this specification work in the context of a user agent's "incognito" mode?

We anticipate that user agents will offer users the ability to grant specific web sites persistent permission to access payment information. This will facilitate user experiences such as “one click” product ordering and automated micropayments.

Recommendation

When operating in an “incognito” mode, we would expect the Payment Request API to remain available; however, we recommend that any such persistent permission be ignored in such a mode (otherwise, websites with such persistent permission would be able to identify users via their payment details). The user agent would still make stored user information available -- similar to how the web browser assists in filling out form information even when incognito; however, such information would be inaccessible to the merchant web site until submitted by the user. Assistance is expected, automation is not.

When operating in incognito mode, it is probably also advisable to take additional steps, possibly at the expense of usability, to frustrate attempts to determine whether the user has registered payment apps that support specific payment methods. For example, always prompting the user when a payment request is made, even if there are no matching payment apps available, may serve such a purpose. Note, however, that this would need careful consideration, as web sites might determine from such behavior that the user is browsing in an incognito context.

When the Payment Request API is invoked in an incognito context, we suggest that any web-based payment apps also be invoked in an incognito context. This will generally prevent such sites from accessing any previously-stored information; this, in turn, will require users to either log in to the payment app or re-enter payment instrument details.

The Payment Request API specification should thus include discussion on browser behavior in incognito mode.

3.15 Does this specification persist data to a user's local device?

The overall system anticipates a model in which payment apps are registered with the browser in a persistent fashion. The considerations for such persistent registration will presumably be treated in the document that defines the means of registration (Payment App API).

There is nothing inherent in the Payments Request API or the Basic Card Payment spec that persistently stores origin-scoped data. The user agent may (and probably will) elect to store mediator-supplied data (e.g., email address, shipping address) on behalf of the user. Payment apps are extremely likely to use existing web technologies to store data, but nothing in the payment API requires such storage, nor does it provide additional mechanism for storage.

Recommendation

None.

3.16 Does this specification have a "Security Considerations" and "Privacy Considerations" section?

Neither the Payment Method Identifier nor the Basic Card Payment specifications contain either of these sections.

Recommendation

The Payment Method Identifier specification should at least point to a URI spec (e.g., RFC 3986) and its security considerations section.

This checklist analysis has indicated a number of privacy and security issues that pertain to the Basic Card Payment spec, and they should be incorporated into the document.

The Payment Request API document does contain privacy and security sections, but they are much smaller than will ultimately be required. They should be updated to reflect the issues discussed in this checklist analysis.

3.17 Does this specification allow downgrading default security characteristics?

We do not believe that it does. Users of the Payment Request API are required to be in a secure context.

Recommendation

At the moment, the Payment Request API indicates the need for operating from a secure context only by means of WebIDL declaration and a somewhat terse note. We recommend that the document include clearer, implementor-targeted prose indicating this restriction.

Although the payment apps specification has not yet been taken up by the Working Group, we also expect that it will require payment apps to run in secure contexts. Information can be transferred between these two contexts using the Payment Request API, but such transfer will be between secure contexts, and (we presume) policed by the user agent.

Additional Notes

Although not explicitly listed in the questionnaire, dealing with financial data does present some unique challenges. In particular, the greatest risk that isn't covered in the current documents or in the preceding checklist is that of payment provider phishing attacks. For example, a web site may register itself as capable of handling the "Basic Card" payment time; and, in fact, may provide valid information to the merchant site. However, if malicious, this site may also exfiltrate the credit card information for its own unauthorized use. The potential risks of installation of payment applications must be very carefully explained to the user.