Hospitality Phishing Cluster uses Personalized Hotel Reservations and Interactive Card Verification

By Ivan Khamenka

Published: 2026-11-06 · Archived: 2026-11-10 02:00:20 UTC

Two templates, one kit

Booking.com template

The first template closely replicates the Booking.com reservation workflow, using the familiar blue branding, property details, stay dates, pricing, cancellation information, and a multi-step checkout process.

Figure 1: Booking.com-themed phishing page displaying reservation information and a multi-step checkout process.

Pages were already populated with what appeared to be reservation-specific information: guest’s name, email address, telephone number, property, check-in and check-out dates, and the price of the stay. This gives the page the appearance of an existing reservation that simply needs to be completed or verified.

The pages also included familiar Booking.com-style language such as refund information, free cancellation messaging, and prompts asking who the reservation is for. In several examples, the property name, photograph, and location were also displayed prominently alongside the guest details.

Behind the Booking.com branding, the JavaScript collects both personal and payment information. The payment stage captures the cardholder's name, card number, expiration date, security code, currency, and payment amount before submitting the data to the phishing backend.

The page also performs validation before accepting the card. It requires a 16-digit number beginning with 4 or 5, rejects obvious test patterns, and performs a Luhn checksum before allowing the victim to continue.

This does not confirm that the card is active or funded, but it filters obviously invalid submissions and suggests the kit is likely designed to collect plausible payment card data rather than simply record whatever is entered.

The Booking.com-branded template also includes a built-in support panel and a verification workflow that continues after the initial card submission, allowing the page to remain interactive rather than ending with a static confirmation screen.

A recurring SHA-256 hash of the phishing kit's `handler.js` file, `8d53c01dc39167d563277c26624f494158d5c24df2a62dedb2357518fe9608b0`, linked many of these deployments.

“Best Hotel Reservation” template

The second template uses a more generic hotel reservation design rather than directly impersonating Booking.com.

Figure 2: Generic Best Hotel Reservation phishing template.

The “Best Hotel Reservation” variant used `handler.js` SHA-256 `7f673442816990817f081064c58b2d232b4e93e356d2a9583250fd0eacc6fb0c`, which appeared in 161 scans at the time of writing.

A small brown logo at the top of the page reads “Best Hotel Reservation” with the words “Travel Company” underneath. The remainder of the page focuses on the individual hotel, stay dates, reservation amount, guest information, and a final payment step.

A reverse-image search of the logo identified the same branding on a Facebook page called “Best Hotel Reservation.” The page uses the same brown circular logo and describes itself as a hotel and travel-related page. Netcraft found no evidence connecting the Facebook page or its operators to the phishing activity. The logo may simply have been copied from publicly available material.



Visually, this template is quite different from the Booking.com version. It uses a neutral brown-and-white design and gives greater prominence to the individual property, making the page appear to belong to an independent booking intermediary or travel company. The branding changes, but the engine underneath does not.

The JavaScript from the two templates contains the same payment card collection, validation logic, backend endpoint structure, interactive verification framework, support chat, session handling, and polling behavior.

The clearest template-specific difference appears at the beginning of the scripts.

The Booking.com version uses:

While the Best Hotel Reservation version uses:

The remainder of the core workflow is effectively unchanged.

Three endpoints support the phishing workflow

Both templates communicate with the same relative backend paths:

Each endpoint has a different role. The **notify** endpoint reports where the victim is in the workflow. The page sends a “details” event when it loads, and a “payment” event when the victim proceeds to the card-entry stage.

The **log** endpoint receives the guest and payment card information. The **page** endpoint supports continued interaction with the backend. It is used to request new instructions, submit challenge responses, send support-chat content, and receive messages or verification states.

Because these paths are relative, the browser sends the data back to the same domain hosting the phishing page. The client-side code observed by the victim therefore reveals the first collection point, but not where the backend stores or forwards the information afterward.

Page tracks victims from the moment it loads

The phishing workflow does not begin only when the victim submits a card. At the bottom of the script, the page immediately reports that the victim has reached the “details” stage and begins polling the backend:

When the victim moves on to the card-entry stage, the page displays the final step and reports a “payment” event:

Requests throughout the workflow contain two identifiers:

These identifiers are included in stage notifications, data submissions, support messages, and backend polling requests. This structure allows the backend to associate activity with an individual victim or session and tailor responses to each visitor.

The card form is only the beginning

After the victim submits payment information, the page can continue presenting additional prompts. This is driven by the polling that began when the page loaded, which checks the page endpoint every three seconds:

Responses from the backend can contain an `action_tds` object. The browser reads that object and uses it to decide what the victim should see next.

The backend can provide:

- A custom message
- An image
- One or two field labels
- A single-field challenge
- A dual-field challenge
- A yes/no confirmation prompt
- A custom error
- A reset instruction
- A redirect to another URL

The wording and labels do not need to be hardcoded into the page. They can be supplied dynamically by the backend. For example, the backend can issue a redirect using a server-supplied destination:

This gives the kit a much more flexible workflow than a static card-harvesting form. The same page can adapt as the interaction progresses.

A dedicated balance-themed state

One of the supported states is named `UP_BALANCE`.

When this state is selected, the page places a backend-supplied message into an element specifically associated with a balance amount. This indicates that the kit can display a balance-themed prompt or value. However, the JavaScript alone does not show the wording of the corresponding page, establish that the victim is asked to provide an account balance, or confirm that a balance is checked against a real financial institution.

The prefix “tds” appears throughout the verification framework and may refer to 3-D Secure (3DS), an authentication protocol used in payment card transactions to help prevent fraud. The available client-side code does not confirm that the phishing backend is connected to a genuine issuer challenge or successfully relays real 3-D Secure prompts. It does, however, show that the kit is designed to imitate or accommodate additional payment-verification steps.

Personalized reservations strengthen the lure

The reservation data is one of the most important parts of the cluster. The pages do not just state that a booking exists. They present a detailed version of that booking, including the property, dates, price, guest identity, email address, and telephone number.

This gives the interaction a sense of continuity. A victim who recognizes their hotel and travel dates may be more likely to trust a request framed as a final verification step, particularly when the page warns that leaving or failing to complete the process could affect the reservation.

The source of the information has not been identified. The personalization does not, by itself, prove that Booking.com, a hotel, or another reservation provider was compromised. The information could originate from compromised hospitality accounts, exposed booking data, earlier phishing activity, another third-party source, or a mixture of genuine and fabricated records.

Adaptable across countries and languages

Many of the observed pages feature Japanese hotels, display prices in yen, and include reservation details associated with travel in Japan. Other examples reference European properties and use EUR pricing.

The JavaScript supports a broad range of languages, including Japanese, Chinese, Korean, Thai, Vietnamese, English, German, French, Spanish, Italian, Portuguese, Arabic, Hebrew, and several others.

The page attempts to select a language and region using the visitor's browser preferences. It also updates the country-code selector and translates page text accordingly. The observed activity is heavily weighted toward

Japanese accommodation, but the kit is designed for much wider geographic use.

Infrastructure and clustering

The phishing pages use several domain-naming conventions rather than relying on direct Booking.com typosquatted domains.

Observed names include combinations of hospitality-related terms such as:

- guest
- rooms
- home
- stay
- reserv
- advanced
- hotel

This approach places most of the impersonation in the page content rather than in the hostname. A domain only needs to appear vaguely relevant to travel or accommodation before the victim is shown the more convincing reservation interface.

Because the Booking.com and Best Hotel Reservation templates share a script structure, minor template-specific changes, such as the translation-object name, can produce different hashes even when the underlying functionality remains the same.

The cluster is better identified through a combination of:

- JavaScript response hashes
- `/include/handler.js`
- `/api/request/log/`
- `/api/request/page/`
- `/api/request/notify/`
- `action_tds`
- `tds_up_balance_amount`
- `support_msg_agent`
- Shared validation and polling logic

Many of the domains were recently registered and placed behind Cloudflare. This limits direct visibility into the origin infrastructure and allows the operators to rotate public-facing domains without changing the underlying kit.

Continued development of the phishing kit

As of Oct. 20, newer versions of the phishing kit had appeared, retaining the same core card harvesting and interactive verification workflow.

Updated versions were observed across both visual branches. The newer Booking.com build, tracked by `handler.js` SHA-256 `a0a71b6a11108d566d1cb3e2d6642ba3ed8c13743363645630c15c8270698f8e` appeared in roughly 100 scans, while the newer Reservation Manager build, tracked by SHA-256 `36f8636b45f8cdf5b053e83f8fbb993bcd0c3dd92393143a918b8c60b7c774ef`, appeared in a further 120 scans.

Figure 4: Updated Reservation Manager template observed in newer deployments of the phishing kit.

Together, the newer variants account for roughly 220 observed scans at the time of writing. Both add a dedicated "**PAYPAL**" state, backend-supplied PayPal email handling, browser-language telemetry, and more granular confirmation states.

The appearance of these changes across both the Booking.com and generic reservation branches, suggests the operators are actively developing the shared framework rather than maintaining a single static template. Additional payment methods or verification workflows are likely to follow as the kit evolves.

Source: https://www.netcraft.com/blog/hospitality-phishing-cluster-uses-personalized-hotel-reservations-and-interactive-card-verification