TENTTI 17.10.2017 TURVALLINEN OHJELMOINTI, TIE-30600

Marko Helenius

Tentissä ei saa käyttää oheismateriaalia eikä laskinta. Vastaa selkeästi.

Vastaa seuraaviin kohtiin turvallisen ohjelmoinnin periaatteiden mukaisesti (6 pistettä/tehtävä):

1. Tarkastele oheisia ohjelmien osia. Mitä haavoittuvuuksia tai ongelmia niissä on, mitä niistä voi seurata ja miksi? Miten korjaisit? (2 pistettä/kohta)

a)

```
TAG_RE = re.compile(r'<(.*?)(\s|>)')  # matches the start of an html tag


def _SanitizeTag(t):
 """Sanitizes a single html tag.

  This does both a 'whitelist' for
  the allowed tags and a 'blacklist' for the disallowed attributes.

 Args:
   t: a tag to sanitize.

 Returns:
   a safe tag.
 """
 allowed_tags = [
     'a', 'b', 'big', 'br', 'center', 'code', 'em', 'h1', 'h2', 'h3',
     'h4', 'h5', 'h6', 'hr', 'i', 'img', 'li', 'ol', 'p', 's', 'small',
     'span', 'strong', 'table', 'td', 'tr', 'u', 'ul',
 ]
 disallowed_attributes = [
     'onblur', 'onchange', 'onclick', 'ondblclick', 'onfocus',
     'onkeydown', 'onkeypress', 'onkeyup', 'onload', 'onmousedown',
     'onmousemove', 'onmouseout', 'onmouseup', 'onreset',
     'onselect', 'onsubmit', 'onunload', 'onmouseover'
 ]

 # Extract the tag name and make sure it's allowed.
 if t.startswith('</'):
   return t
 m = TAG_RE.match(t)
  if m is None:
    return t
 tag_name = m.group(1)
 if tag_name not in allowed_tags:
   t = t[:m.start(1)] + 'blocked' + t[m.end(1):]

 # This is a bit heavy handed but we want to be sure we don't
 # allow any to get through.
 for a in disallowed_attributes:
    t = t.replace(a, 'blocked')
  return t
```


Käännä!