

Tietoverkkorikollisuus

Rikoskomisario Teppo Viljanen

```
144
145 /**
146  * @param {*} item
147  * @param {Comparator} [comparator]
148  * @return {Heap}
149  */
150 remove(item, comparator = this.compare) {
151   // Find number of items to remove.
152   const numberOfItemsToRemove = this.find(item, comparator).length;
153
154   for (let iteration = 0; iteration < numberOfItemsToRemove; iteration++) {
155     // We need to find item index to remove each time after removal
156     // indices are being changed after each heapify process.
157     const indexToRemove = this.find(item, comparator).pop();
158
159     // If we need to remove last child in the heap then just remove it.
160     // There is no need to heapify the heap afterwards.
161     if (indexToRemove === (this.heapContainer.length - 1)) {
162       this.heapContainer.pop();
163     } else {
164       // Move last element in heap to the vacant (removed) position.
165       this.heapContainer[indexToRemove] = this.heapContainer.pop();
166     }
167   }
168 }
169
170 // Init helper variables
171 const distances = {};
172 const visitedVertices = {};
173 const previousVertices = {};
174 const queue = new PriorityQueue();
175
176 // Init all distances to infinity
177 // any of the vertices
178 graph.getAllVertices().forEach((vertex) => {
179   distances[vertex] = Infinity;
180   previousVertices[vertex] = null;
181 });
182
183 // We are already at the start vertex
184 distances[startVertex] = 0;
185
186 // Init vertices queue.
187 queue.add(startVertex, distances[startVertex]);
188
189 // Iterate over the priority queue until it is empty
190 while (!queue.isEmpty()) {
191   const vertex = queue.poll();
192   // Visit all neighbors of the current vertex
193   graph.getNeighbors(vertex).forEach((neighbor) => {
194     // Calculate the distance to the neighbor
195     const distanceToNeighbor = distances[vertex] + graph.getEdgeWeight(vertex, neighbor);
196     // If the distance to the neighbor is less than the current distance, update it
197     if (distanceToNeighbor < distances[neighbor]) {
198       distances[neighbor] = distanceToNeighbor;
199       previousVertices[neighbor] = vertex;
200     }
201     // Add the neighbor to the queue
202     queue.add(neighbor, distanceToNeighbor);
203   });
204 }
205
206 // Return the shortest path
207 return previousVertices;
208 }
```

SISÄLLYS

1. Yleistä
2. Palvelunestohyökkäykset
3. Haittaohjelmat
4. Tietomurrot
5. Tietojenkalastelusta
6. TVA-petoksista
7. Suojautuminen

Yleistä

- Ihmisten elämä ja palvelut ovat siirtyneet kasvavassa määrin verkkoon
- Riippuvuus verkkoliikenteestä
- Erialaisten älylaitteiden (IoT) lisääntyminen luo uusia uhkia
- Tekoäly tuo mukanaan uhkia ja mahdollisuuksia
- Järjestelmien ja ohjelmistojen haavoittuvuudet
- Pilvipalvelut
- Vakoilu

Palvelunestohyökkäykset

- DDoS = Distributed Denial of Service
- Hajautettu hyökkäys internetissä toimivaa palvelua kohtaan
- Toteutetaan siten, että järjestelmää tai verkkosivustoa ylikuormitetaan valtavalla määrällä verkkoliikennettä, jonka seurauksena aiheutuu viivettä palvelussa tai pahimmillaan koko palvelun kaatuminen.
- Ddos as a Service
 - Kyberrikollisuuden teollistuminen
 - Hyökkäykset saatavissa helposti ns. kuluttajapalveluna

Haittaohjelmat

- Suunniteltu vahingoittamaan laitetta, varastamaan tietoja tai häiritsemään järjestelmiä.
- Esimerkiksi virukset, madot, troijalaiset, vakoilu- ja mainosohjelmat
- Kohteina tietokoneet, puhelimet ja muut älylaitteet
- Lisää osaksi botti-verkkoa
- Kiristyshaittaohjelmat oma alalaji
- Käytetään myös huijausviesteissä uhkauksen perusteena
- Tekoälyn rooli nousemassa koko ajan

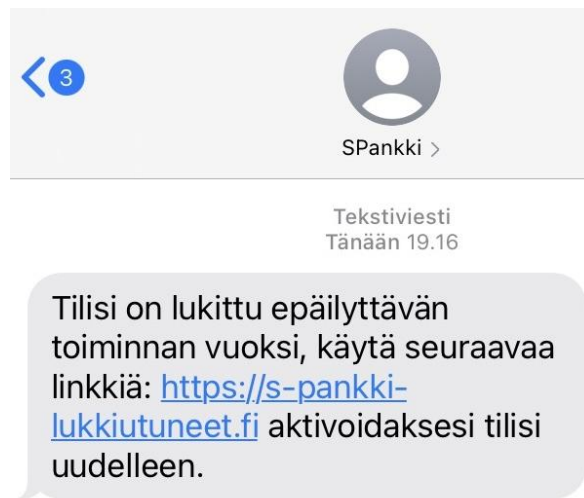
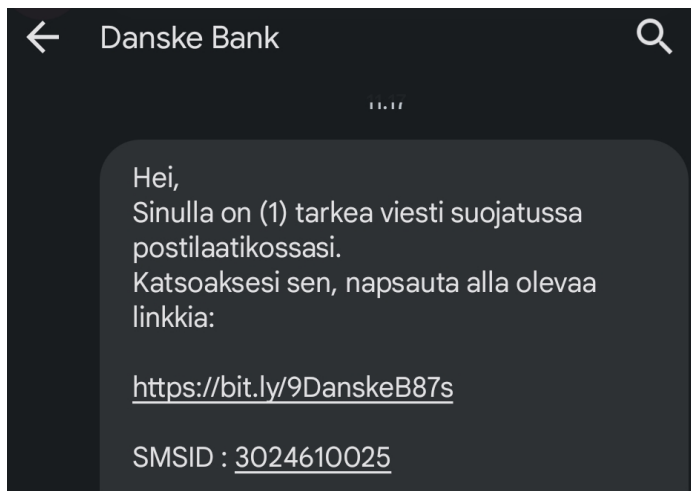
Tietomurrot

- Yleisimpinä sosiaalisen median tilit ja M365
- Toimii yleensä esitekona mahdollistamaan muuta toimintaa
 - Roskapostituslistat
 - Tietojenkalastelu
 - Valelaskutus
 - Kiristäminen
- Tietomurrolla saatuja tietoja voidaan myydä eteenpäin dark netissä tai keskustelupalstoilla, esimerkiksi Telegramissa.

Tietojenkallastelusta

- Pyritään saamaan maksukorttien tietoja, pankkitunnuksia, salasanoja ja käyttäjätunnuksia sekä mobiilivarmenteita.
- Naamioidaan esim. pankista, postista tai viranomaiselta tulevaksi chat-, sms- tai sähköpostiviestiksi. Uusimpana tekemuotona Teams-puhelut.
- Hakukonemainonta
- Kilpailut ja kyselyt
- Myyntialustojen kuriiripalvelut
- Lähtökohtaisesti pyritään saamaan tietoja petosten toteuttamiseksi.
- Noudattelevat reaali maailman tapahtumia.

Tietojenkallastelusta



Maksuvälineet ja tietojenkalastelu

- Palvelintilan hankinta
- Verkkosoitteen (domain) rekisteröinti
- Valeverkkosivuston rakentaminen
- Tietojenkalastelupaneelin käyttöönotto
- Uhrien erehdyttäminen valesivustolle
- Maksuvälineen oikeudeton käyttö
- Rahanpesun toteuttaminen

TVA-petoksista

- Vuonna 2024 yli 32.000 rikosilmoitusta
- Rikollisten saama rikoshyöty yli 84 milj. euroa
- Suurimmat rikosvahingot
 - Tietojenkalastelupetokset 37,8 milj. euroa
 - Sijoituspetokset 30,8 milj. euroa
- Muita tekotapoja
 - Kauppapaikkapetokset
 - Dokumentti- ja rakkaushuijaukset
 - Toimitusjohtajahuijaukset
 - Asiakaspalveluhuijaukset
- Pyritään estämään yhteistyössä pankkien, operaattoreiden ja Traficomin kanssa.

Tietojenkalastelupetoksista

- Keskeisimmät tekijät huijauksissa
 - Murretut sähköposti- ja sosiaalisen median tilit
 - Verkkosoite
 - Linkit
 - Liitetiedostot
 - Hakukoneen tulokset
 - Etäkäyttöyhteydet
- Viime aikoina käytetty eri tekijöiden yhdistelmiä
 - Esim. soitto, viesti, valesivusto, haittaohjelma

Tietojenkalastelupetoksista



Lähde: GCF Global

- [posti.fi](#)
- [posti.fi-app.net](#)
- [identify.nordea.com](#)
- [identify-nordea.com](#)

Suojautuminen

- Vahvat salasanat
- MFA
- Linkit ja tunnistautuminen / maksaminen
- Liian hyvää ollakseen totta ja / tai kiire, niin yleensä petos
- Kriittinen suhtautuminen
- Verko-osoitteen kirjoittaminen itse / selaimen kirjanmerkit
- Varmistusviestien sisällön huolellinen lukeminen
- Mobiilivarmenteen käyttö estää pankkitilille pääsemisen

Kiitos