

SURF SECURE ID

Extra beveiligde toegang tot uw diensten

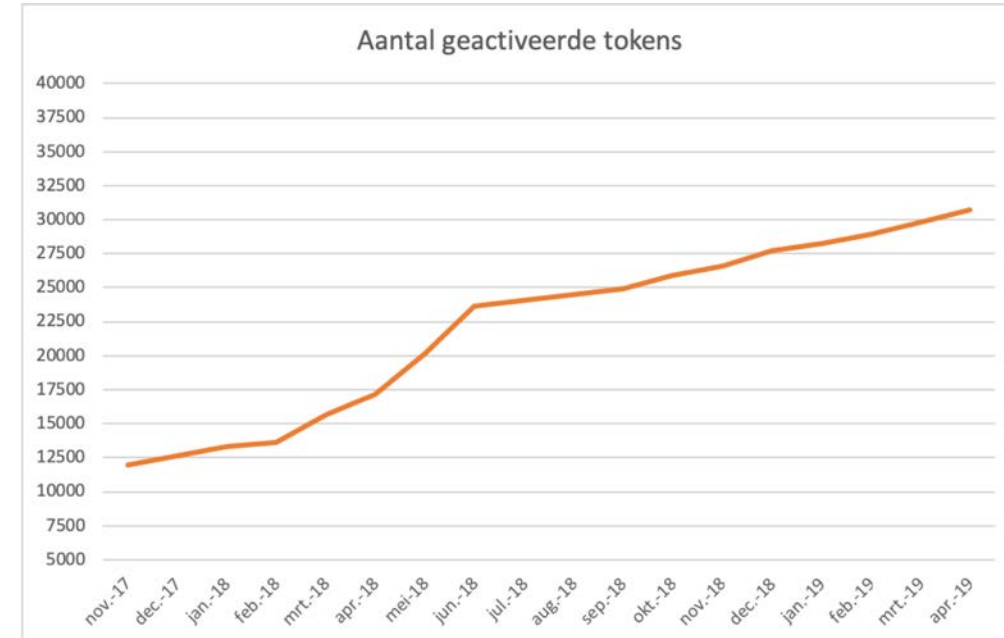
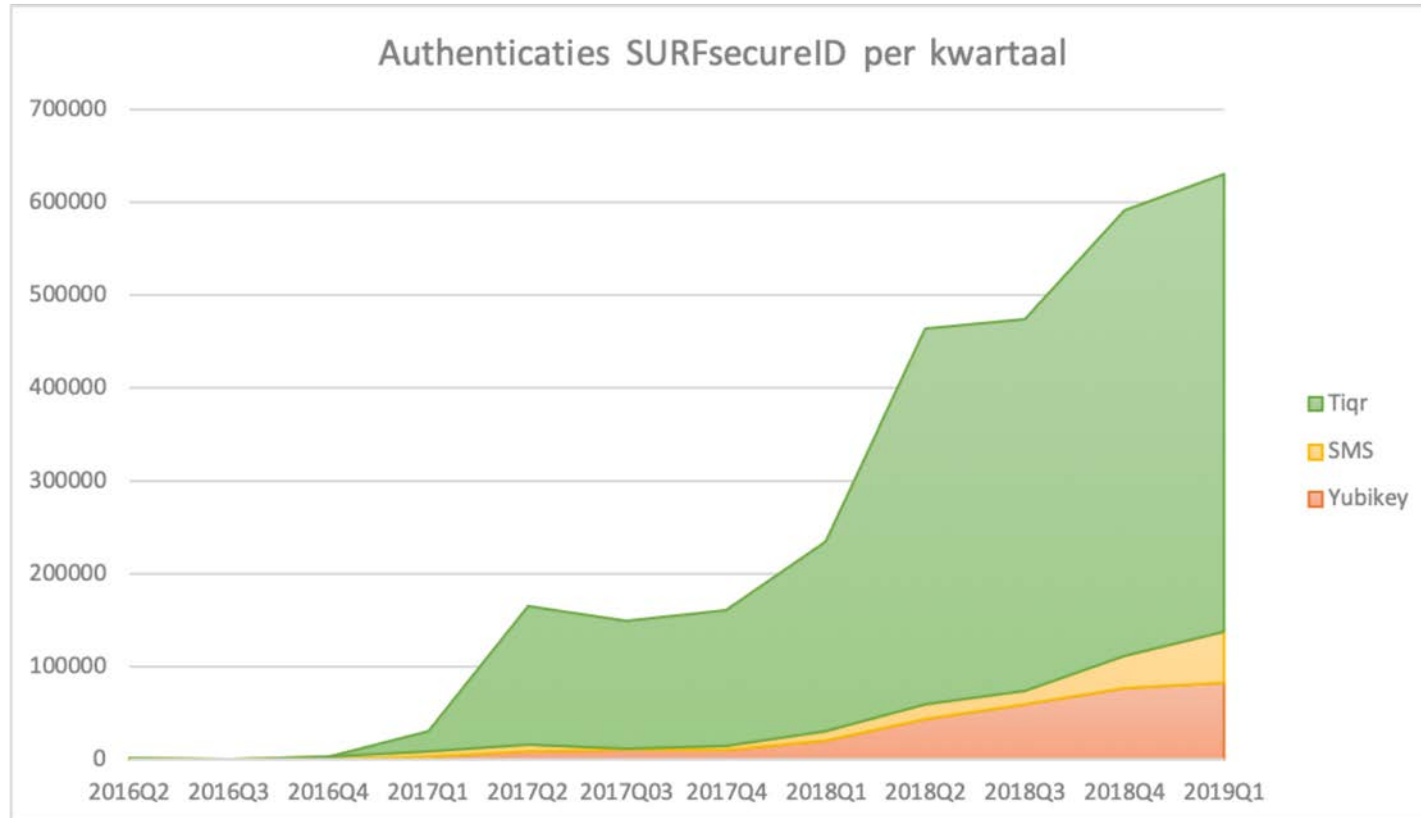


SURF NET

- Extra beveiligde toegang voor alle diensten
- Gebruiker kiest zijn authenticatie middel
- Hoog betrouwbaarheidsniveau met out-of-the-box registratieproces
- Inzicht en controle op uitgedeelde middelen
- SURFsecureID deel je met alle instellingen
- SURFsecureID is open source en gebaseerd op open standaarden



Gebruik SURFsecureID



Aangesloten instellingen en diensten





Terugblik

- Herpositionering van SURFconext Sterke Authenticatie naar SURFsecureID
- Tariefaanpassing
- Ondersteuning meerdere tokens per gebruiker
- Optioneel uitzetten email verificatie bij token registratie
- Onderhoud en vernieuwing tiqr server en ondergebracht bij onze reguliere ontwikkelpartij
- Nieuwe tiqr apps met ondersteuning van vingerafdruk / faceID
- SURFsecureID als MFA middel voor Azure AD
- Aantal storingen met tiqr en DDoS aanval



Update remote vetting project

- Hoe kan iemand online geïdentificeerd worden met vergelijkbare betrouwbaarheidsniveaus als huidig proces?
- Innovalor onderzoekt 3 methoden: iDIN, ReadID (NFC), IRMA
- NFC en IRMA leveren attributen uit het BRP
- Matching met instellingsidentiteit
- Welke processtappen moeten we opnemen
- Idee om eerder geregistreerde SSID middelen (Tiqr, Yubikey, etc) ook te kunnen gebruiken voor remote vetting
- Voorlopige planning; eind april ontwerp klaar. Vervolgens PoC uitvoeren en daarna pilots.

Vooruitblik

- Autorisatie model voor registratie tussen instellingen
- Voorziening voor registreren op afstand (remote vetting)
- Niet geverifieerde tokens ondersteunen
- Integratie in SURFconext en OpenID Connect ondersteuning
- FIDO2 WebAuthn
- Azure MFA als token voor SURFsecureID
- SURFsecureID voor SURF diensten

```
31 def __init__(self, settings):
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'fingerprints.txt'), 'w')
39         self.file.seek(0)
40         self.fingerprints.update(fingerprint(request) for request in requests)
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool('SURFsecureID', False)
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```



Jullie input!

www.menti.com

code: 71 86 61





Peter Clijsters

peter.clijsters@surfnet.nl

<https://www.surfsecureid.nl>

<https://twitter.com/SURFnet>

02
STERKE
AUTHENTICATIE

01
WACHTWOORD

