The FASHIONCLEFT Protocol

[redacted]

STDP, S32354

“go tiny-url?q=p5atn”

Definition: FASHIONCLEFT

- *TAO/DNT protocol used by implants to exfiltrate collected network packets to the Common Data Receptor (CDR).*
- Provides support for:
  - Metadata Authentication/Integrity + AntiReplay + Encryption
  - Data Encryption
  - Uses 1024-bit RSA, 128-bit RC6
- Based on DNT standards:
  - FOGYNULL (Exfil Protocol)
  - FUNNELAPS (Exfil Data Format)
  - SHELLGREY (Exfil Metadata Format)

How To Exfiltrate IP Packets

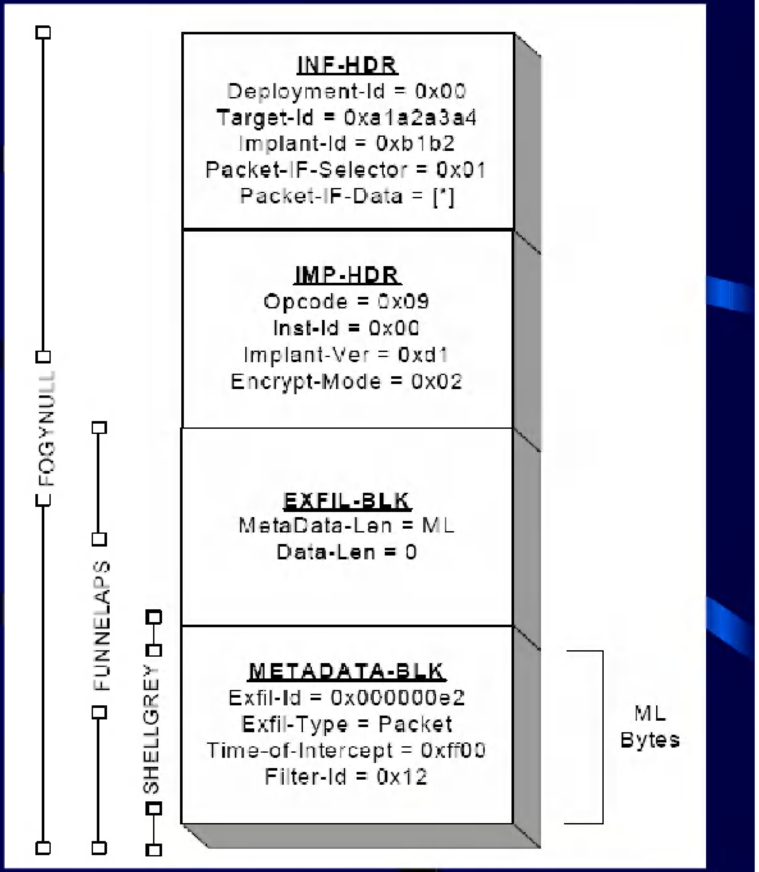
1. Make a copy of the captured packet.
2. Modify packet IP destination address.
3. Modify other protocol fields (IP, UDP, TCP) as needed to bypass firewalls and tag packets for ID.
4. Optionally encrypt/munge Transport layer payload.
5. Send modified Data Packet (DP) to new destination.

Receiver: Needs Metadata

- Metadata explains how to:
  1. Identify an exfil packet and the implant source.
  2. Recover original IP destination address.
  3. Recover other original protocol fields (IP, UDP, TCP).
  4. Contains Key to decrypt/unmunge transport layer payload.
- Metadata sent in a Session Announcement (SA)
  - SAs use IP/UDP or IP/TCP sent to an IP/port.
  - Multiple copies of SA sent to mitigate dropped SA packets.
- Receiver is dynamically configured with:
  - SA IP/ports, Infrastructure & Implant Private Keys

Session Announcement Format

- IP Header
- TCP or UDP Header
- SA Payload
  - Infrastructure Header (128 bytes)
    - RSA Encrypted w/ Infrastructure Public Key
    - Contains SHA-1(INF-HDR), ID
      - ID = Deployment-Id + Target-Id + Implant-Id
  - Implant Header (128 bytes)
    - RSA Encrypted w/ ID’s Public Key
    - Contains SHA-1(IMP-HDR)
    - 128-bit CV, MI, and CRC-16 checksum for Exfil/Metadata Block
  - Exfil/Metadata Block (variable)
    - RC6 Encrypted w/ CV & MI
- Minimum packet length = 344 bytes



INF-HDR
Deployment-Id = 0x00
Target-Id = 0xa1a2a3a4
Implant-Id = 0xb1b2
Packet-IF-Selector = 0x01
Packet-IF-Data = [*]

IMP-HDR
Opcode = 0x09
Inst-Id = 0x00
Implant-Ver = 0xd1
Encrypt-Mode = 0x02

EXFIL-BLK
MetaData-Len = ML
Data-Len = 0

METADATA-BLK
Exfil-Id = 0x000000e2
Exfil-Type = Packet
Time-of-Intercept = 0xff00
Filter-Id = 0x12

Session Announcement Processing

1. Look for SAs at IP/port that are at least 344 bytes long.
   - `ppf::api::KeywordCriterion(IP.dstAddr, IP.dstPort)`
   - (Easy/quick initial check)
1. RSA Decrypt INF-HDR w/ Infrastructure Private Key.
   - Authenticate w/ SHA-1
   - (Slow secondary check; can’t withstand much non-SA traffic on IP/port)
1. RSA Decrypt IMP-HDR w/ ID’s Private Key.
   - Authenticate w/ SHA-1
1. RC6 Decrypt Exfil/Metadata w/ CV and MI
   - Perform CRC-16 integrity check.
1. Extract Metadata and create Data Packet (DP) filter rule.
   - Metadata contains either 5-tuples or pattern/mask/offset that match DPs
   - `ppf::api::KeywordCriterion(5-tuple or pattern/mask/offset)`

Data Packet Processing

1. Identify an exfil packet that matches DP filter rule.
2. Modify to original IP destination address.
3. Modify to original protocol fields (IP, UDP, TCP).
4. Decrypt/unmunge transport layer payload.
   - Have now recovered the original captured packet.

1. Associate metadata with recovered packet.
   - Implant CASN, Turmoil link CASN
1. Perform protocol specific processing.
   - Reinject? Bundle?
   - Need option to force packets to be “strongly selected”.

Questions?

Supplemental Material

FASHIONCLEFT & Turmoil

- Adding FASHIONCLEFT capability to Turmoil supports these missions:
  - VPN
    - Provide IKE/ISAKMP key exchanges obtained from unique TAO accesses to the VPN Attack Orchestrator.
  - VoIP
    - Create new high bandwidth exfiltration path to Turmoil for streaming VoIP to overcome limited CDR bandwidth.
  - Others
    - Automatic exfil path discovery?
    - Etc…

Library Reuse: CDR □ PPF

- TAO Common Data Receptor
  - Access Control Point (ACP, C)
  - SURPASSPIN Inner/Outer (SP-in, SP-out, Java)
  - libftsk (FOGYNULL Technique Software Kit, C)
- Turmoil Packet Processing Framework (C++)
  - Atomic Event Generator (AEG)
  - Stateful Event Generator (SEG)
  - Event Filter (EF)
  - Packet-to-Packet Transform Engine (TE)

ACP "Equivalent API"

- Cache::setTimeWindow
- Cache::setSize
- Cache::getInfo
- Cache::clear
- Cache::enableArchiving
- Cache::disableArchiving
- Cache::getArchivingStatus

- SaFilter::set
- SaFilter::delete
- SaFilter::getList
- SaFilter::clearList

- DpFilter::set
  - Create DpFilter & check cache for match
- DpFilter::delete
- DpFilter::getList
- DpFilter::clearList

- Acp::getStatus
  - Cache::getArchivingStatus
  - SaFilter::getList
  - DpFilter::getList
  - Cache::getInfo

- Acp::checkRawPacket
  - (packet processing callback)
  - cache the packet
    - if cacheFull
      - archiveIfEnabled
      - warnIfInCacheTimeWindow
  - find/process all matching DpFilters
  - find/process unique matching SaFilter

To be Determined

- Tasking & Monitoring of Turmoil
  - Add/Delete/Query tasking (JMS ITx?)
  - Add/Delete/Query other processing/config options (MBean?)
  - Tasking/configuration persistence
  - Processing metrics & logging
  - Should tasking use CDR .icf files? (Implant Config Files)
  - Should Turmoil interface w/ PUZZLECUBE? (TAO tasking database)
- Protocol Processing
  - Metadata: Implant CASN + Turmoil link CASN
  - Reinject? Packet Bundling?
    - VoIP, VPN, etc.
  - Force Strong Selection Option: On/Off
  - Turmoil 30-sec DFCE vs. CDR 15-minute packet cache

Current CDR Tasking

- FLASHHANDLE Mission Manager (FMM)
  - – Provides tasking to CDR/SURPASSPIN
- FMM Server
  - – Reads configuration information from the PUZZLECUBE database
  - – Allows the operator to add/change tasking
    - (including generating implant encryption keys)
- Tasking changes are:
  - – Sent back to PUZZLECUBE via JDBC messages
  - – Published via JMS messages to SURPASSPIN
    - SURPASSPIN stores the tasking in a persistent POJO cache.

Implant Configuration File (.icf)

[redacted]

```
# (4843) HAMMERCHANT
ICF_NAME       4843.a1b20000.00000113.21Mar2007
ICF DTG        Wed Mar 21 18:12:33 2007
ICF INFO       (4843) HAMMERCHANT FOR TARGET ID
               0xa1b20000
IMPLANT_ID     0x4843
IMPLANT_VER    1
TARGET_ID      0xa1b20000
DEPLOYMENT_ID  0x00000113
TARGET_CN      HAMMERCHANT_BatonRouge
TARGET_IP      172.32.6.113
TARGET_HOST    BatonRouge
#
# IMPLANT_LP[1-9]  [Tunnel-Id:]ip-address[:port(s)]
# Tunnel-Id:       2-Fashioncleft
IMPLANT_LP1      2:68.1.1.178:12000
IMPLANT_LP2      2:68.1.1.178:12001
IMPLANT_RC6_CV1  e3d3ae0a b341ade1 4dce30e0 77861acc
```

```
#
IMPLANT RSA INF
RSANAME Infrastructure_Key_E.rsa
RSAINF0 Wed Aug 25 10:17:29 2004, rsagenkey v2.0
RSASIZE 1024
RSAMOD 32
  0xe420b8d5, 0x47673b7a, 0xaf4c39a1, 0xc704d5ba,
  […7 lines deleted…]
RSAMU 33
  0xed5692b1, 0x449323bb, 0xed7653e5, 0xcd9feb5e,
  […7 lines deleted…]
  0x00000002
RSAPRIV 32
  0x63c5f12b, 0xd1b85426, 0x4f5a681c, 0x68be4748,
  […7 lines deleted…]
RSAPUB 32
  0x00000003, 0x00000000, 0x00000000, 0x00000000,
  […7 lines deleted…]
#
IMPLANT_RSA_IMP
RSANAME
  (4843)_HAMMERCHANT_at_HAMMERCHANT_(a1b20000/00000113)
[…same format as IMPLANT_RSA_INF…]
```

Packet Cache Options

- CDR uses a 15 minute packet cache.
  - SAs are sent multiple times per session and the cache is searched for matching DPs to mitigate dropped SAs.
- Simple Cache:
  - Use existing Turmoil cache (Delay Flow Control Engine).
- Large Cache:
  - Create a large cache that allows a 15 minute delay.
- Options:
  - Start with Simple cache and see if we miss too many DPs. If problems then implement Large cache.
  - Start with a Large cache and see if we can keep up with data rate & memory requirements. If problems then scale back to Simple cache.

Simple Packet Cache

- The hardware LightDelay provides a 30 second cache.
- The software XFSPF provides a 2 second cache.

- Pros:
  1. No problems with buffering data since Turmoil does it automatically.
  2. No work required to implement cache.
- Cons:
  1. Cache is much smaller than 15 minute (900 seconds = 30x 30) CDR requirement.
  2. Cache delay is further reduced by unspecified latency to register new DP filters after receipt of SA.
  3. Many DPs would be ignored if SA is missed/delayed.
     – Possibly "mitigated" by sending multiple SA copies in first 30 (or 2) seconds of exfil.

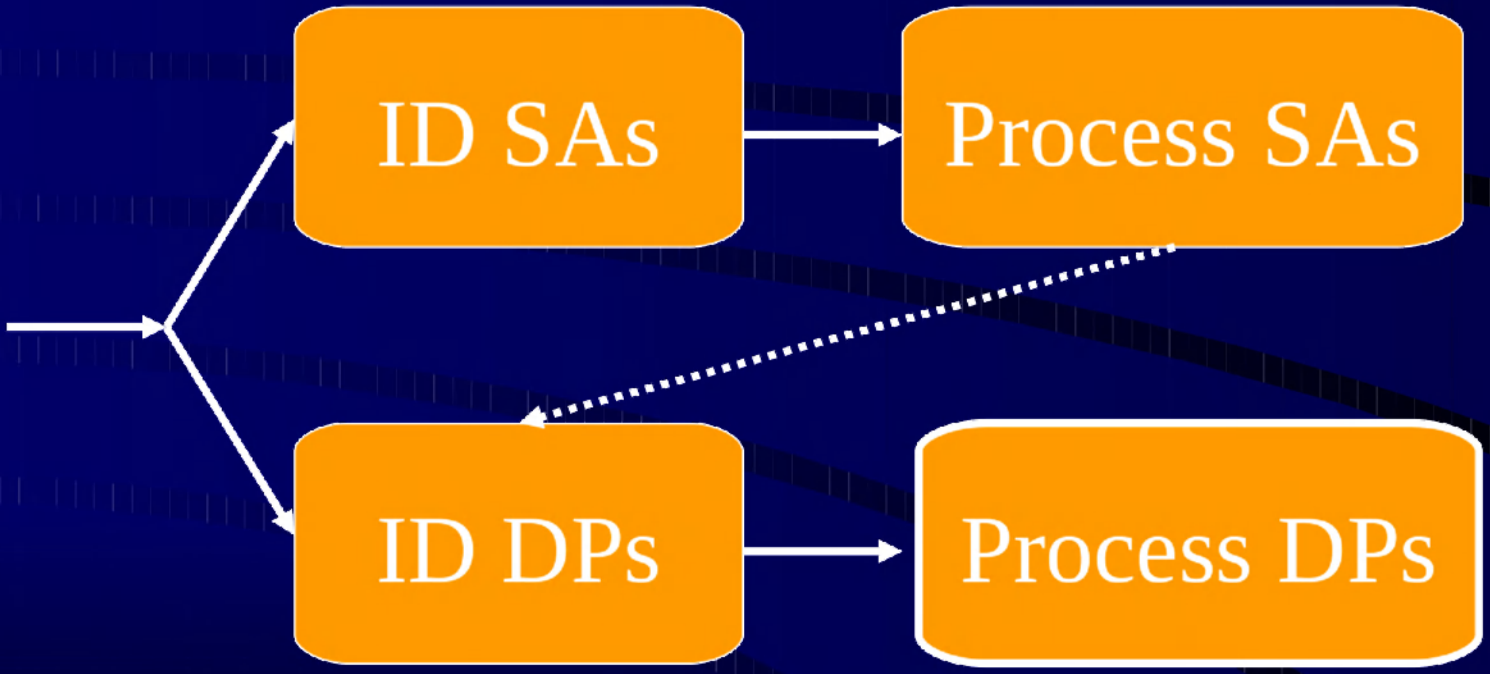
Large Packet Cache

- Implement large 15-minute packet cache within AEG.

- Pros:
  1. Meets CDR cache requirement.
  2. Most/all DPs should be processed even if initial SA is missed/delayed.

- Cons:
  1. Violates normal Turmoil architecture. May not be possible/feasible to implement a large cache at typical Turmoil rates.
  2. Requires caching all IP packets sent to "CDR IP address", then manually searching for DP hits instead of letting the PPF search packets.
  3. Time/effort required to implement.

SA/DP ID & Processing


ID SAs
Process SAs
ID DPs
Process DPs