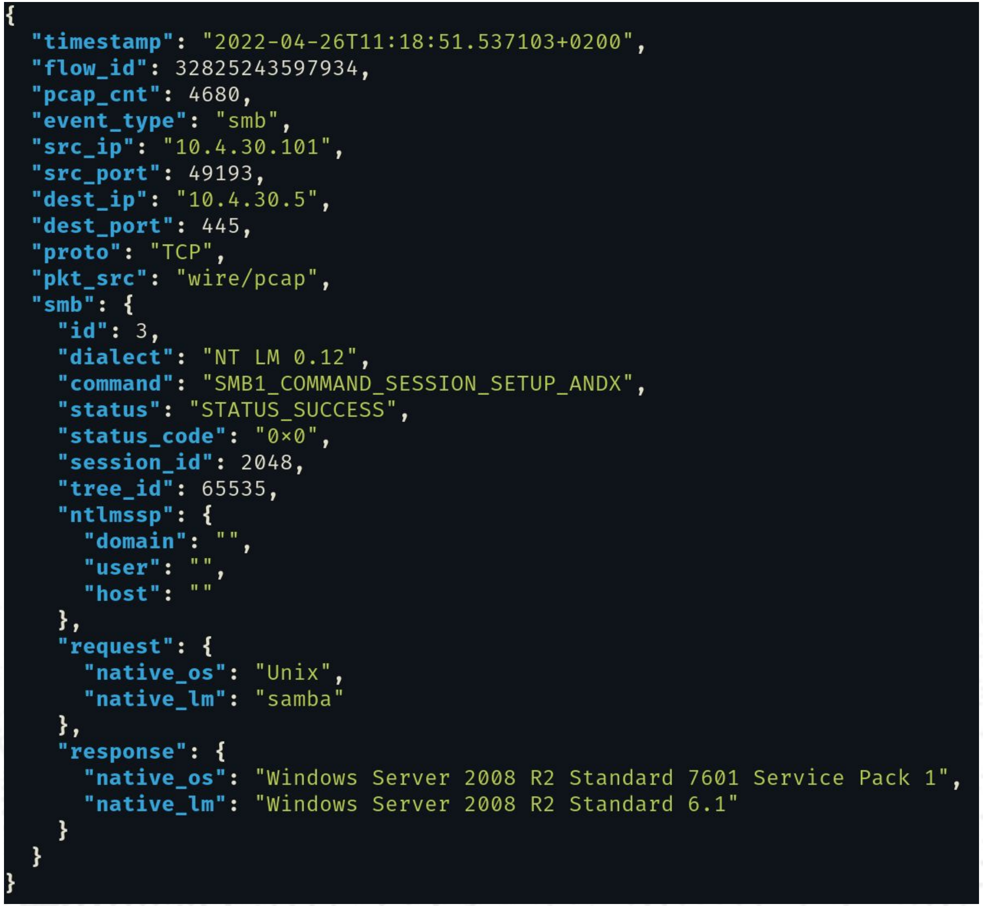
STAMVS NETWORKS

Laziness is a serious job

Why you should complain more.

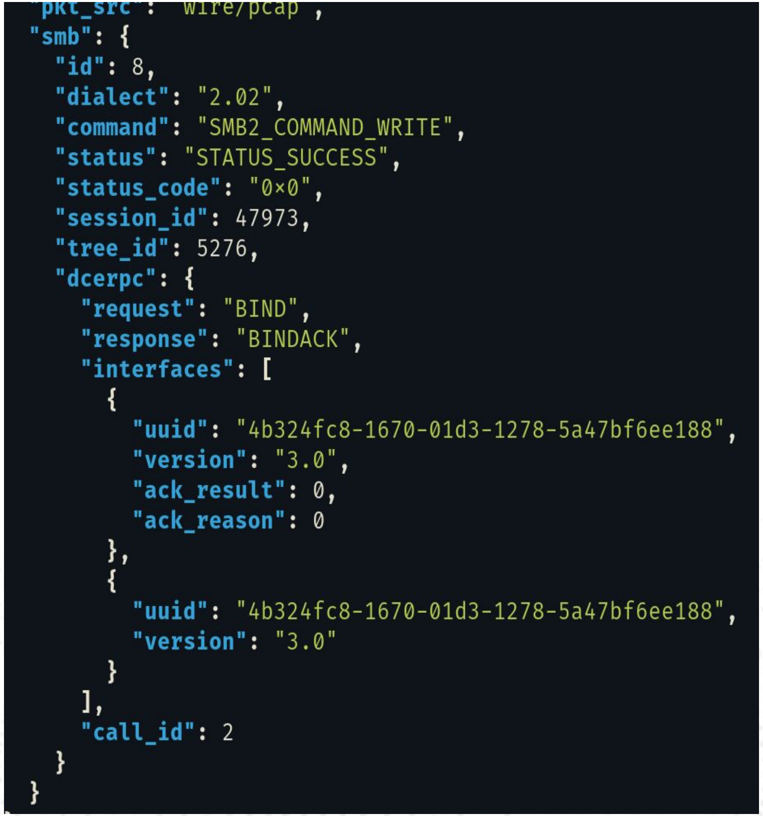
Suricata SMB logging

- Suricata NSM capabilities
  - Log all transactions
  - Independently of alerts
- Available for:
  - Http, dns, ftp, tls
  - Dhcp, smb, krb5, nfs
  - Dnp3, enip, modbus, mqtt
  - …

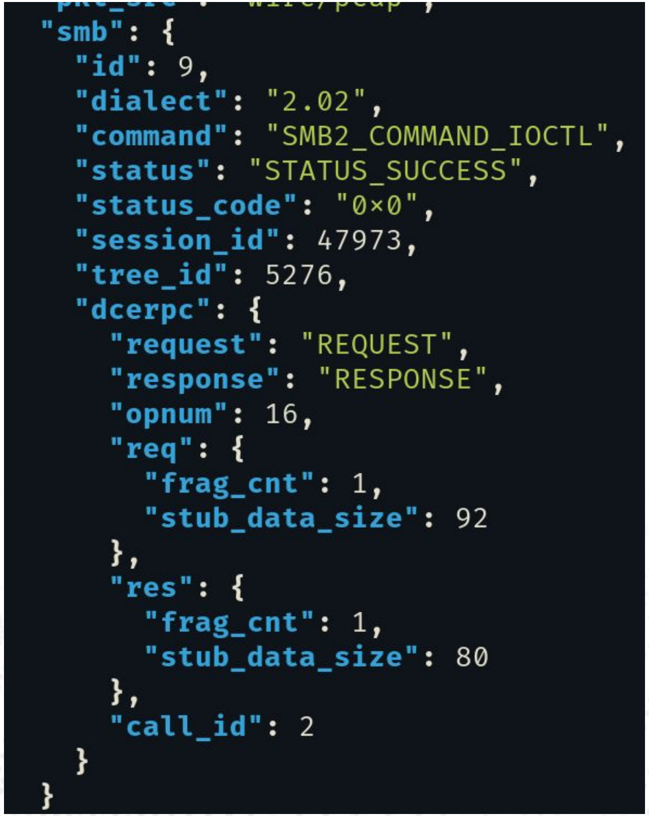


```
{
  "timestamp": "2022-04-26T11:18:51.537103+0200",
  "flow_id": 32825243597934,
  "pcap_cnt": 4680,
  "event_type": "smb",
  "src_ip": "10.4.30.101",
  "src_port": 49193,
  "dest_ip": "10.4.30.5",
  "dest_port": 445,
  "proto": "TCP",
  "pkt_src": "wire/pcap",
  "smb": {
    "id": 3,
    "dialect": "NT LM 0.12",
    "command": "SMB1_COMMAND_SESSION_SETUP_ANDX",
    "status": "STATUS_SUCCESS",
    "status_code": "0x0",
    "session_id": 2048,
    "tree_id": 65535,
    "ntlmssp": {
      "domain": "",
      "user": "",
      "host": ""
    },
    "request": {
      "native_os": "Unix",
      "native_lm": "samba"
    },
    "response": {
      "native_os": "Windows Server 2008 R2 Standard 7601 Service Pack 1",
      "native_lm": "Windows Server 2008 R2 Standard 6.1"
    }
  }
}
```

SMB DCERPC events



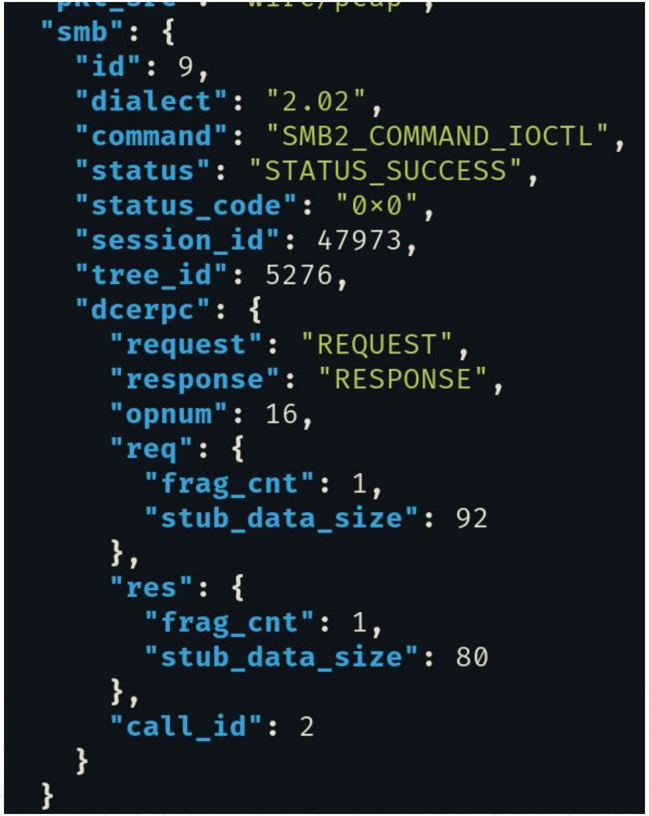
```
"pkt_src": "wire/pcap",
"smb": {
  "id": 8,
  "dialect": "2.02",
  "command": "SMB2_COMMAND_WRITE",
  "status": "STATUS_SUCCESS",
  "status_code": "0x0",
  "session_id": 47973,
  "tree_id": 5276,
  "dcerpc": {
    "request": "BIND",
    "response": "BINDACK",
    "interfaces": [
      {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0",
        "ack_result": 0,
        "ack_reason": 0
      },
      {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0"
      }
    ],
    "call_id": 2
  }
}
```



```
"pkt_src": "wire/pcap",
"smb": {
  "id": 9,
  "dialect": "2.02",
  "command": "SMB2_COMMAND_IOCTL",
  "status": "STATUS_SUCCESS",
  "status_code": "0x0",
  "session_id": 47973,
  "tree_id": 5276,
  "dcerpc": {
    "request": "REQUEST",
    "response": "RESPONSE",
    "opnum": 16,
    "req": {
      "frag_cnt": 1,
      "stub_data_size": 92
    },
    "res": {
      "frag_cnt": 1,
      "stub_data_size": 80
    },
    "call_id": 2
  }
}
```

What is wrong here ?

- DCERPC call is defined via
  - UUID
  - Operation number
- Screenshot example:
  - UUID: 4b324fc8-1670-01d3-1278-5a47bf6ee188
    - Name: srvsvc
  - Opnum: 16
    - Endpoint: NetrShareGetInfo
- Consequence:
  - Need correlation of 2 events
  - Really complicated on some environments



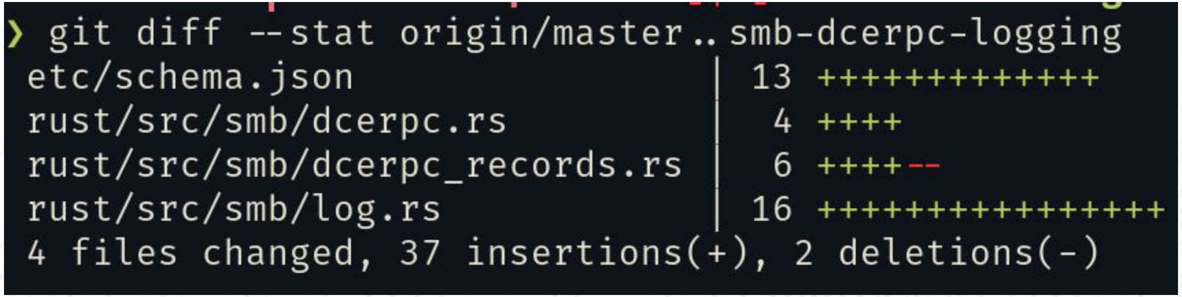
```
  "pkt_src": "wire/pcap",
  "smb": {
    "id": 9,
    "dialect": "2.02",
    "command": "SMB2_COMMAND_IOCTL",
    "status": "STATUS_SUCCESS",
    "status_code": "0x0",
    "session_id": 47973,
    "tree_id": 5276,
    "dcerpc": {
      "request": "REQUEST",
      "response": "RESPONSE",
      "opnum": 16,
      "req": {
        "frag_cnt": 1,
        "stub_data_size": 92
      },
      "res": {
        "frag_cnt": 1,
        "stub_data_size": 80
      },
      "call_id": 2
    }
  }
```

- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-srvs/b2c2b995-8c42-4502-b7d7-646d1b295c5b

Laziness can require skills

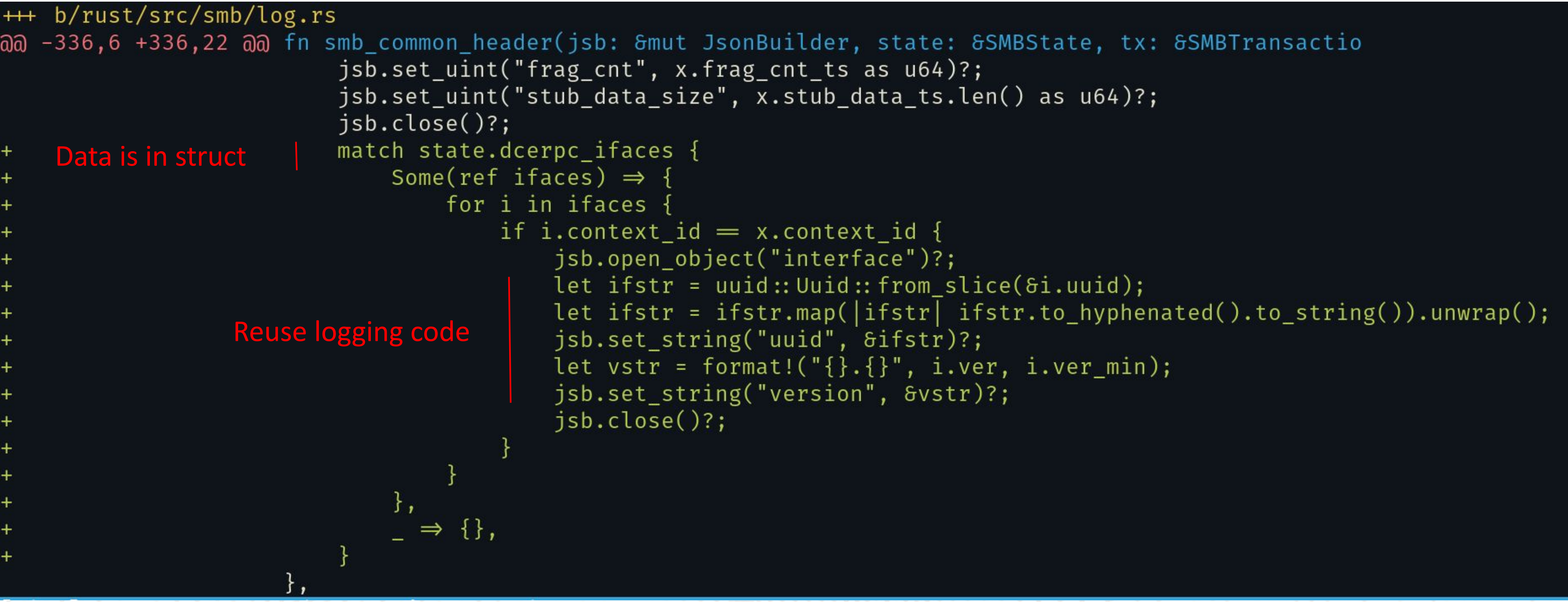
Let’s get 2 infos in the last events

- We need to hack:
  - Protocol logging
  - Potentially protocol parser
- Patch doing the implementation is small:



```
❯ git diff --stat origin/master..smb-dcerpc-logging
 etc/schema.json                   | 13 +++++++++++++
 rust/src/smb/dcerpc.rs            |  4 ++++
 rust/src/smb/dcerpc_records.rs    |  6 ++++--
 rust/src/smb/log.rs               | 16 ++++++++++++++++
 4 files changed, 37 insertions(+), 2 deletions(-)
```

Logging



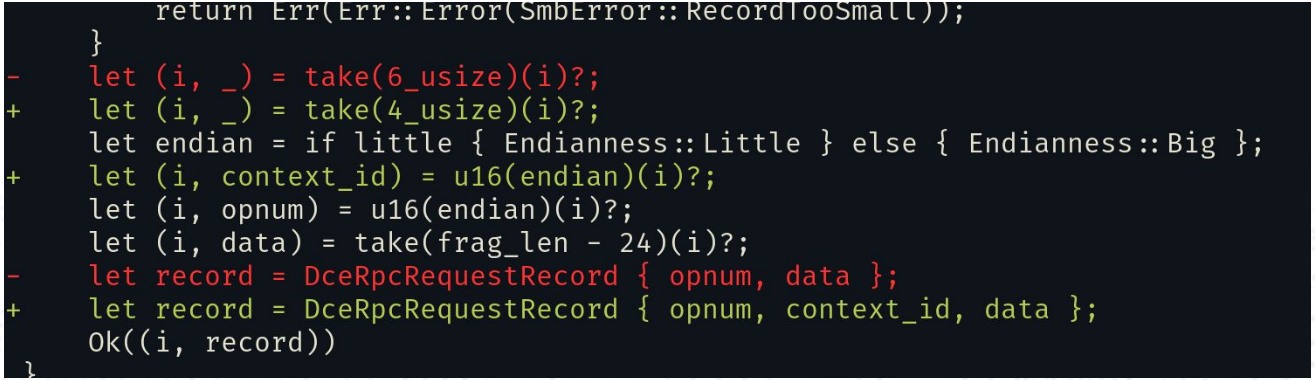
```
+++ b/rust/src/smb/log.rs
@@ -336,6 +336,22 @@ fn smb_common_header(jsb: &mut JsonBuilder, state: &SMBState, tx: &SMBTransactio
             jsb.set_uint("frag_cnt", x.frag_cnt_ts as u64)?;
             jsb.set_uint("stub_data_size", x.stub_data_ts.len() as u64)?;
             jsb.close()?;
+            match state.dcerpc_ifaces {
+              Some(ref ifaces) => {
+                for i in ifaces {
+                  if i.context_id == x.context_id {
+                    jsb.open_object("interface")?;
+                    let ifstr = uuid::Uuid::from_slice(&i.uuid);
+                    let ifstr = ifstr.map(|ifstr| ifstr.to_hyphenated().to_string()).unwrap();
+                    jsb.set_string("uuid", &ifstr)?;
+                    let vstr = format!("{}.{}", i.ver, i.ver_min);
+                    jsb.set_string("version", &vstr)?;
+                    jsb.close()?;
+                  }
+                }
+              },
+              _ => {},
+            }
          },
```

Data is in struct

Reuse logging code

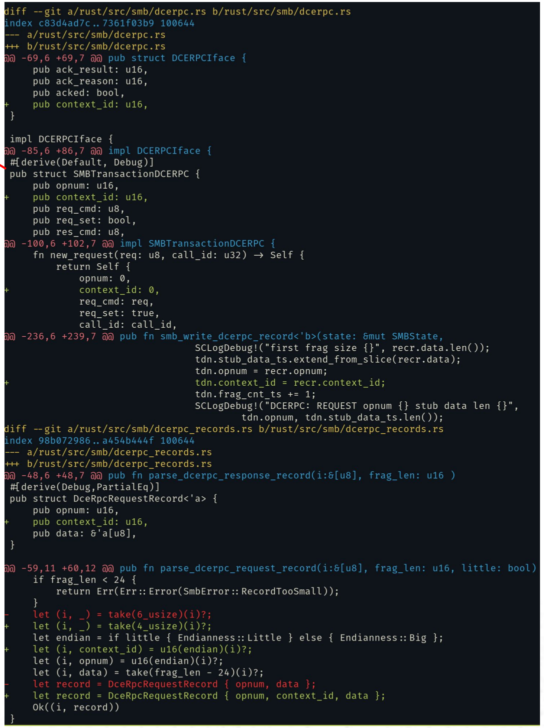
Parsing context_id

- Context_id was not parsed
- Adding a field is trivial
    - Thanks to NOM usage
    - https://github.com/Geal/nom



```
        return Err(Err::Error(SmbError::RecordTooSmall));
    }
-   let (i, _) = take(6_usize)(i)?;
+   let (i, _) = take(4_usize)(i)?;
    let endian = if little { Endianness::Little } else { Endianness::Big };
+   let (i, context_id) = u16(endian)(i)?;
    let (i, opnum) = u16(endian)(i)?;
    let (i, data) = take(frag_len - 24)(i)?;
-   let record = DceRpcRequestRecord { opnum, data };
+   let record = DceRpcRequestRecord { opnum, context_id, data };
    Ok((i, record))
}
```

Full patch

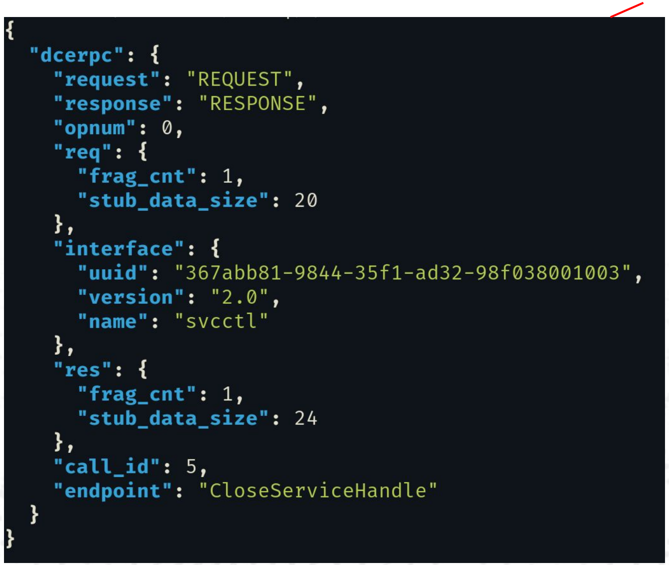


```
diff --git a/rust/src/smb/dcerpc.rs b/rust/src/smb/dcerpc.rs
index c83d4ad7c..7361f03b9 100644
--- a/rust/src/smb/dcerpc.rs
+++ b/rust/src/smb/dcerpc.rs
@@ -69,6 +69,7 @@ pub struct DCERPCIface {
     pub ack_result: u16,
     pub ack_reason: u16,
     pub acked: bool,
+    pub context_id: u16,
 }

 impl DCERPCIface {
@@ -85,6 +86,7 @@ impl DCERPCIface {
 #[derive(Default, Debug)]
 pub struct SMBTransactionDCERPC {
     pub opnum: u16,
+    pub context_id: u16,
     pub req_cmd: u8,
     pub req_set: bool,
     pub res_cmd: u8,
@@ -100,6 +102,7 @@ impl SMBTransactionDCERPC {
     fn new_request(req: u8, call_id: u32) -> Self {
         return Self {
             opnum: 0,
+            context_id: 0,
             req_cmd: req,
             req_set: true,
             call_id: call_id,
@@ -236,6 +239,7 @@ pub fn smb_write_dcerpc_record<'b>(state: &mut SMBState,
                                 SCLogDebug!("first frag size {}", recr.data.len());
                                 tdn.stub_data_ts.extend_from_slice(recr.data);
                                 tdn.opnum = recr.opnum;
+                                tdn.context_id = recr.context_id;
                                 tdn.frag_cnt_ts += 1;
                                 SCLogDebug!("DCERPC: REQUEST opnum {} stub data len {}",
                                         tdn.opnum, tdn.stub_data_ts.len());
diff --git a/rust/src/smb/dcerpc_records.rs b/rust/src/smb/dcerpc_records.rs
index 98b072986..a454b444f 100644
--- a/rust/src/smb/dcerpc_records.rs
+++ b/rust/src/smb/dcerpc_records.rs
@@ -48,6 +48,7 @@ pub fn parse_dcerpc_response_record(i:&[u8], frag_len: u16 )
 #[derive(Debug,PartialEq)]
 pub struct DceRpcRequestRecord<'a> {
     pub opnum: u16,
+    pub context_id: u16,
     pub data: &'a[u8],
 }

@@ -59,11 +60,12 @@ pub fn parse_dcerpc_request_record(i:&[u8], frag_len: u16, little: bool)
     if frag_len < 24 {
         return Err(Err::Error(SmbError::RecordTooSmall));
     }
-    let (i, _) = take(6_usize)(i)?;
+    let (i, _) = take(4_usize)(i)?;
     let endian = if little { Endianness::Little } else { Endianness::Big };
+    let (i, context_id) = u16(endian)(i)?;
     let (i, opnum) = u16(endian)(i)?;
     let (i, data) = take(frag_len - 24)(i)?;
-    let record = DceRpcRequestRecord { opnum, data };
+    let record = DceRpcRequestRecord { opnum, context_id, data };
     Ok((i, record))
 }
```

Result of modification

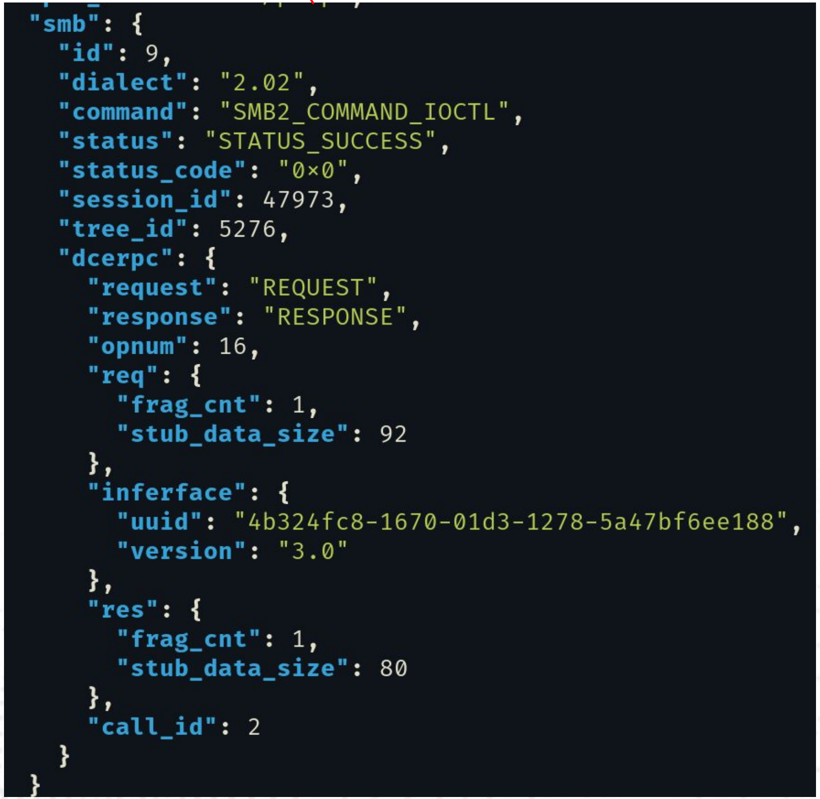
- All in one!
- You can postprocess to get
  - UUID Name
  - Endpoint Name

Post processed



```
{
  "dcerpc": {
    "request": "REQUEST",
    "response": "RESPONSE",
    "opnum": 0,
    "req": {
      "frag_cnt": 1,
      "stub_data_size": 20
    },
    "interface": {
      "uuid": "367abb81-9844-35f1-ad32-98f038001003",
      "version": "2.0",
      "name": "svcctl"
    },
    "res": {
      "frag_cnt": 1,
      "stub_data_size": 24
    },
    "call_id": 5,
    "endpoint": "CloseServiceHandle"
  }
}
```

Suricata event



```
  "smb": {
    "id": 9,
    "dialect": "2.02",
    "command": "SMB2_COMMAND_IOCTL",
    "status": "STATUS_SUCCESS",
    "status_code": "0x0",
    "session_id": 47973,
    "tree_id": 5276,
    "dcerpc": {
      "request": "REQUEST",
      "response": "RESPONSE",
      "opnum": 16,
      "req": {
        "frag_cnt": 1,
        "stub_data_size": 92
      },
      "inferface": {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0"
      },
      "res": {
        "frag_cnt": 1,
        "stub_data_size": 80
      },
      "call_id": 2
    }
  }
```

Conclusion

- COMPLAIN
  - Don’t assume things are like they are and can’t change
    - Best developers do mistake
  - Question the data, question the tool
    - Ask forums
    - Ask the developer
  - If correctly asked questions are not welcomed, switch to another tool
- CONTRIBUTE
  - Give yourself a chance to do the hack yourself
  - Your patch may be just a few copy paste away

Suricata PR: https://github.com/OISF/suricata/pull/7584