Windows Security OSSIR group
13th September 2004

Active Directory network protocols and traffic

Jean-Baptiste Marchand
<Jean-Baptiste.Marchand@hsc.fr>
Agenda

✓ Active Directory network protocols overview
✓ Network traffic analysis with ethereal
✓ Network traffic for each protocol
✓ Active Directory typical scenarios
✓ Other approaches
✓ Conclusion
✓ References
Active Directory network protocols

- Active Directory is based on network protocols
  - Standardized: DNS, LDAP, Kerberos V, SNTP
  - Proprietary: SMB/CIFS, MSRPC
- Use of Internet protocols, embraced and extended by Microsoft
Internet protocols: DNS

- DNS

- Specifications: many RFCs
  - http://www.dns.net/dnsrd/rfc/

- Name resolution service (replaces NetBIOS name resolution used in NT domains)
  - Dynamic DNS entries update
  - GSS-TSIG (RFC 3645)

- Domain services localization
  - SRV DNS records
Internet protocols: LDAP

- LDAP
  - Specifications: see RFC 3377
  - Active Directory is a directory that can be queried using LDAP
    - Ports 389 (TCP and UDP), 636 (LDAPS), 3268 and 3269 (AD Global Catalog)
  - Specific SASL mechanism: GSS-SPNEGO
  - Windows systems also access Active Directory using MSRPC
    - `samr` and `drsuapi` RPC interfaces
  - Sensitive information is encrypted
    - LDAP sessions using TCP port 389, encrypted using GSS-SPNEGO
    - Encrypted MSRPC operations (*packet privacy*)
  - LDAP does not include directory replication standardization
    - Active Directory replication uses MSRPC or SMTP
Internet protocols: Kerberos V

* Kerberos V
  * Network authentication protocol
  * Protocol defined at MIT then standardized at the IETF, widely used in Unix environments
  * Embraced and extended by Microsoft
    * RC4-HMAC cipher, TCP transport, PAC (Privilege Access Certificate), PKINIT, ...
    * Standard interfaces are implemented for compatibility but are not used by native Windows clients
    * Example: kpasswd service (for password changing)
  * Kerberos V has been integrated to Windows services using the SSPI layer
    * SPNEGO, for negotiation between different security packages (NTLM, Kerberos V, Schannel, ...)
Internet protocols: SNTP

✗ SNTP

✗ Simple Network Time Protocol, version 3 (RFC 1769)
✗ Simplified version of the NTP protocol (RFC 1305)
  ✓ same packet format, using UDP port 123
  ✓ less precise than NTP (but enough for Kerberos V)
✗ Synchronization packets are signed
  ✓ usually ignored in SNTP
  ✓ used to authenticate synchronization packets
Proprietary protocols: SMB/CIFS

- SMB/CIFS
  - Windows domains resource sharing protocol
  - Frequently confused with NetBIOS over TCP/IP
  - Used for file and printer sharing
  - Also a possible transport for MSRPC
    - Transport using named pipes (ncacn_np)
  - Active Directory prefers TCP/IP transport, as opposed to NT 4.0
    - SMB transport is still used when a machine is joined to a domain
  - Group Policy: sysvol share
    - gpt.ini, registry.pol, *.adm, GptTmpl.inf files
  - Connection scripts: netlogon share
Proprietary protocols: MSRPC

- MSRPC
  - MS implementation of the DCE RPC standard
  - Active Directory domains are based on key RPC interfaces:
    - lsarpc: LSA access (Local Security Authority)
    - netlogon: network authentication service
    - samr: SAM access (NT 4.0 backward compatibility, works with Active Directory)
    - drsuapi: Active Directory access
  - Active Directory uses TCP transport for these RPC services
    - Portmapper listening on TCP port 135
    - Default ports range for RPC services listening on TCP
      - 1025-5000 (default interval), to be modified with rpccfg
    - Reminder: NT 4.0 was based on RPC services over SMB, over NetBIOS over TCP/IP (TCP port 139)
Network authentication

- Kerberos V is the network authentication protocol used in AD
  - Replaces NTLM
  - Supports mutual authentication
  - Aforementioned network protocols have been modified to support Kerberos
    - SMB/CIFS sessions authentication
    - LDAP sessions authentication
    - MSRPC calls authentication
    - Dynamic DNS updates authentication
  - Kerberos V support was added using a negotiation protocol, SPNEGO (Simple Protected Negotiation Mechanism, RFC 2478)
    - Multiple errors in Microsoft SPNEGO implementation, leading to serious interoperability problems
Network traffic analysis: goals

- Possible goals of network traffic analysis
  - Understanding Active Directory
  - Validating key mechanisms of Active Directory domains
    - Ex 1: Kerberos tickets renewal
    - Ex 2: Group Policy processing
  - Tracking anomalies
Network traffic analysis: methodology

✗ Require access to domain controllers network traffic
  ✗ To capture network traffic
✗ Require a network analyzer supporting aforementionned protocols
  ✗ Recommended network analyzer: ethereal
  ✗ Free software, working on Unix and Windows
  ✗ Support of multiple network protocols, including Windows-oriented protocols (SMB/CIFS and MSRPC)
  ✗ Support of Kerberos tickets decryption
    ✗ On Unix with Heimdal (http://www.pdc.kth.se/heimdal/)
    ✗ http://www.ethereal.com/
Network traffic typology

- Network traffic typology overview
  - Examining observed protocols
    - ethereal Protocol Hierarchy function
  - Examining traffic typology
    - ethereal Conversations function
    - IPv4 conversations: conversations at the IP level
    - TCP, UDP conversation: (IP addresses, ports) (source and destination)
# Protocol Hierarchy function

<table>
<thead>
<tr>
<th>Protocol</th>
<th>% Packets</th>
<th>Packets</th>
<th>Bytes</th>
<th>Mbit/s</th>
<th>End Packets</th>
<th>End Bytes</th>
<th>End Mbit/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame</td>
<td>100.00%</td>
<td>278</td>
<td>73952</td>
<td>0.003</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>Ethernet</td>
<td>100.00%</td>
<td>278</td>
<td>73952</td>
<td>0.003</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>Address Resolution Protocol</td>
<td>2.88%</td>
<td>8</td>
<td>462</td>
<td>0.000</td>
<td>8</td>
<td>462</td>
<td>0.000</td>
</tr>
<tr>
<td>Internet Protocol</td>
<td>97.12%</td>
<td>270</td>
<td>73490</td>
<td>0.003</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>User Datagram Protocol</td>
<td>12.95%</td>
<td>36</td>
<td>20158</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>Domain Name Service</td>
<td>5.04%</td>
<td>14</td>
<td>2123</td>
<td>0.000</td>
<td>14</td>
<td>2123</td>
<td>0.000</td>
</tr>
<tr>
<td>Lightweight Directory Access Protocol</td>
<td>2.16%</td>
<td>6</td>
<td>1315</td>
<td>0.000</td>
<td>3</td>
<td>664</td>
<td>0.000</td>
</tr>
<tr>
<td>Kerberos</td>
<td>5.04%</td>
<td>14</td>
<td>16500</td>
<td>0.001</td>
<td>14</td>
<td>16500</td>
<td>0.001</td>
</tr>
<tr>
<td>Network Time Protocol</td>
<td>0.72%</td>
<td>2</td>
<td>220</td>
<td>0.000</td>
<td>2</td>
<td>220</td>
<td>0.000</td>
</tr>
<tr>
<td>Transmission Control Protocol</td>
<td>82.73%</td>
<td>230</td>
<td>53082</td>
<td>0.002</td>
<td>114</td>
<td>12540</td>
<td>0.001</td>
</tr>
<tr>
<td>DCE RPC</td>
<td>12.95%</td>
<td>36</td>
<td>9223</td>
<td>0.000</td>
<td>12</td>
<td>2823</td>
<td>0.000</td>
</tr>
<tr>
<td>DCE/RPC Endpoint Mapper</td>
<td>2.16%</td>
<td>6</td>
<td>1160</td>
<td>0.000</td>
<td>6</td>
<td>1160</td>
<td>0.000</td>
</tr>
<tr>
<td>Microsoft Network Logon</td>
<td>3.60%</td>
<td>10</td>
<td>3376</td>
<td>0.000</td>
<td>10</td>
<td>3376</td>
<td>0.000</td>
</tr>
<tr>
<td>Microsoft Directory Replication Service</td>
<td>2.86%</td>
<td>8</td>
<td>1864</td>
<td>0.000</td>
<td>8</td>
<td>1864</td>
<td>0.000</td>
</tr>
<tr>
<td>NetBIOS Session Service</td>
<td>11.51%</td>
<td>32</td>
<td>6669</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>SMB (Server Message Block Protocol)</td>
<td>11.51%</td>
<td>32</td>
<td>6669</td>
<td>0.000</td>
<td>32</td>
<td>6669</td>
<td>0.000</td>
</tr>
<tr>
<td>Lightweight Directory Access Protocol</td>
<td>15.83%</td>
<td>44</td>
<td>23807</td>
<td>0.001</td>
<td>42</td>
<td>22481</td>
<td>0.001</td>
</tr>
<tr>
<td>Hypertext Transfer Protocol</td>
<td>0.72%</td>
<td>2</td>
<td>1326</td>
<td>0.000</td>
<td>2</td>
<td>1326</td>
<td>0.000</td>
</tr>
<tr>
<td>Internet Control Message Protocol</td>
<td>1.44%</td>
<td>4</td>
<td>843</td>
<td>0.000</td>
<td>4</td>
<td>843</td>
<td>0.000</td>
</tr>
<tr>
<td>Internet Control Message Protocol</td>
<td>1.44%</td>
<td>4</td>
<td>250</td>
<td>0.000</td>
<td>4</td>
<td>250</td>
<td>0.000</td>
</tr>
</tbody>
</table>
TCP conversations

<table>
<thead>
<tr>
<th>Address A</th>
<th>Port A</th>
<th>Address B</th>
<th>Port B</th>
<th>Packets</th>
<th>Bytes</th>
<th>Packets A-&gt;B</th>
<th>Bytes A-&gt;B</th>
<th>Packets A&lt;-B</th>
<th>Bytes A&lt;-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.70.106.144</td>
<td>1046</td>
<td>192.70.106.151</td>
<td>80</td>
<td>11</td>
<td>1249</td>
<td>5</td>
<td>625</td>
<td>6</td>
<td>624</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1029</td>
<td>192.70.106.151</td>
<td>135</td>
<td>15</td>
<td>1604</td>
<td>9</td>
<td>908</td>
<td>6</td>
<td>696</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1052</td>
<td>192.70.106.151</td>
<td>135</td>
<td>12</td>
<td>1040</td>
<td>7</td>
<td>638</td>
<td>5</td>
<td>402</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1056</td>
<td>192.70.106.151</td>
<td>389</td>
<td>23</td>
<td>7166</td>
<td>13</td>
<td>3157</td>
<td>10</td>
<td>4009</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1043</td>
<td>192.70.106.151</td>
<td>389</td>
<td>20</td>
<td>7815</td>
<td>12</td>
<td>3096</td>
<td>8</td>
<td>4719</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1045</td>
<td>192.70.106.151</td>
<td>389</td>
<td>15</td>
<td>3103</td>
<td>9</td>
<td>2240</td>
<td>6</td>
<td>863</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1057</td>
<td>192.70.106.151</td>
<td>389</td>
<td>12</td>
<td>3891</td>
<td>7</td>
<td>2019</td>
<td>5</td>
<td>1872</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1058</td>
<td>192.70.106.151</td>
<td>389</td>
<td>12</td>
<td>3823</td>
<td>7</td>
<td>1974</td>
<td>5</td>
<td>1849</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1059</td>
<td>192.70.106.151</td>
<td>389</td>
<td>12</td>
<td>3869</td>
<td>7</td>
<td>2019</td>
<td>5</td>
<td>1850</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1035</td>
<td>192.70.106.151</td>
<td>445</td>
<td>44</td>
<td>8835</td>
<td>25</td>
<td>5700</td>
<td>19</td>
<td>3135</td>
</tr>
<tr>
<td><strong>192.70.106.144</strong></td>
<td><strong>1039</strong></td>
<td><strong>192.70.106.151</strong></td>
<td><strong>1025</strong></td>
<td><strong>22</strong></td>
<td><strong>5685</strong></td>
<td><strong>12</strong></td>
<td><strong>4078</strong></td>
<td><strong>10</strong></td>
<td><strong>1607</strong></td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1031</td>
<td>192.70.106.151</td>
<td>1025</td>
<td>18</td>
<td>3702</td>
<td>11</td>
<td>2324</td>
<td>7</td>
<td>1378</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1030</td>
<td>192.70.106.151</td>
<td>1025</td>
<td>14</td>
<td>1300</td>
<td>8</td>
<td>828</td>
<td>6</td>
<td>472</td>
</tr>
</tbody>
</table>
# UDP Conversations

<table>
<thead>
<tr>
<th>Address A</th>
<th>Port A</th>
<th>Address B</th>
<th>Port B</th>
<th>Packets</th>
<th>Bytes</th>
<th>Packets A-&gt;B</th>
<th>Bytes A-&gt;B</th>
<th>Packets A&lt;-B</th>
<th>Bytes A&lt;-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.70.106.144</td>
<td>1026</td>
<td>192.70.106.151</td>
<td>53</td>
<td>4</td>
<td>758</td>
<td>2</td>
<td>244</td>
<td>2</td>
<td>514</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1047</td>
<td>192.70.106.151</td>
<td>53</td>
<td>2</td>
<td>252</td>
<td>1</td>
<td>83</td>
<td>1</td>
<td>169</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1048</td>
<td>192.70.106.151</td>
<td>53</td>
<td>2</td>
<td>252</td>
<td>1</td>
<td>126</td>
<td>1</td>
<td>126</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1049</td>
<td>192.70.106.151</td>
<td>53</td>
<td>2</td>
<td>251</td>
<td>1</td>
<td>87</td>
<td>1</td>
<td>164</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1050</td>
<td>192.70.106.151</td>
<td>53</td>
<td>2</td>
<td>231</td>
<td>1</td>
<td>171</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1051</td>
<td>192.70.106.151</td>
<td>53</td>
<td>2</td>
<td>379</td>
<td>1</td>
<td>122</td>
<td>1</td>
<td>257</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1036</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>1728</td>
<td>1</td>
<td>361</td>
<td>1</td>
<td>1367</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1037</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>2666</td>
<td>1</td>
<td>1346</td>
<td>1</td>
<td>1320</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1038</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>2645</td>
<td>1</td>
<td>1339</td>
<td>1</td>
<td>1306</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1040</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>2666</td>
<td>1</td>
<td>1346</td>
<td>1</td>
<td>1320</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1041</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>2600</td>
<td>1</td>
<td>1324</td>
<td>1</td>
<td>1276</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1042</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>2702</td>
<td>1</td>
<td>1364</td>
<td>1</td>
<td>1356</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1051</td>
<td>192.70.106.151</td>
<td>88</td>
<td>2</td>
<td>1475</td>
<td>1</td>
<td>1335</td>
<td>1</td>
<td>140</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>123</td>
<td>192.70.106.151</td>
<td>123</td>
<td>2</td>
<td>220</td>
<td>1</td>
<td>110</td>
<td>1</td>
<td>110</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1028</td>
<td>192.70.106.151</td>
<td>389</td>
<td>2</td>
<td>467</td>
<td>1</td>
<td>250</td>
<td>1</td>
<td>217</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1034</td>
<td>192.70.106.151</td>
<td>389</td>
<td>2</td>
<td>424</td>
<td>1</td>
<td>207</td>
<td>1</td>
<td>217</td>
</tr>
<tr>
<td>192.70.106.144</td>
<td>1054</td>
<td>192.70.106.151</td>
<td>389</td>
<td>2</td>
<td>424</td>
<td>1</td>
<td>207</td>
<td>1</td>
<td>217</td>
</tr>
</tbody>
</table>
Network traffic filtering

- ethereal supports display filters
- Most of ethereal dissectors give access to filterable fields, corresponding to data fields decoded in data frames
- Displayed frames filtering can be specified using any filterable fields
- Apply as filter and Prepare a filter functions
Display filters examples

Display filters for Active Directory protocols

- `smb`: SMB sessions
- `ldap & udp`: CLDAP traffic
- `ldap & tcp`: LDAP traffic
- `dcerpc`: MSRPC traffic
- `kerberos & udp`: Kerberos exchanges (UDP port 88)
- `kerberos.msg.type == 10`: AS-REQ Kerberos messages
- `smb & kerberos, ldap & kerberos, dcerpc & kerberos`: Kerberos authentication frames inside SMB, LDAP and MSRPC (AP-REQ and AP-REP messages)
  - Equivalent to: `kerberos & tcp`
## Kerberos authentication: SMB, MSRPC, LDAP

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>55</td>
<td>2004-07-15 11:18:56,957848</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>SMB</td>
<td>Session Setup AndX Request</td>
</tr>
<tr>
<td>57</td>
<td>2004-07-15 11:18:57,016509</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>SMB</td>
<td>Session Setup AndX Response</td>
</tr>
<tr>
<td>75</td>
<td>2004-07-15 11:18:57,927875</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>DCERPC</td>
<td>Bind; call_id: 1 UUID: DRSUAPI</td>
</tr>
<tr>
<td>77</td>
<td>2004-07-15 11:18:57,958316</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>DCERPC</td>
<td>Bind_ack; call_id: 1 accept max&gt;</td>
</tr>
<tr>
<td>78</td>
<td>2004-07-15 11:18:57,960004</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>DCERPC</td>
<td>Alter_context; call_id: 1 UUID: I</td>
</tr>
<tr>
<td>104</td>
<td>2004-07-15 11:18:58,521506</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=3 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>105</td>
<td>2004-07-15 11:18:58,599533</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=3 Bind Result</td>
</tr>
<tr>
<td>112</td>
<td>2004-07-15 11:18:58,667594</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=7 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>113</td>
<td>2004-07-15 11:18:58,749381</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=7 Bind Result</td>
</tr>
<tr>
<td>217</td>
<td>2004-07-15 11:20:00,033894</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=3 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>218</td>
<td>2004-07-15 11:20:00,116670</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=3 Bind Result</td>
</tr>
<tr>
<td>230</td>
<td>2004-07-15 11:20:00,238930</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=10 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>231</td>
<td>2004-07-15 11:20:00,324893</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=10 Bind Result</td>
</tr>
<tr>
<td>243</td>
<td>2004-07-15 11:20:00,381976</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=15 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>244</td>
<td>2004-07-15 11:20:00,474899</td>
<td>192.70.106.151</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=15 Bind Result</td>
</tr>
<tr>
<td>255</td>
<td>2004-07-15 11:20:00,504705</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=20 Bind Request, DN=(null)</td>
</tr>
<tr>
<td>256</td>
<td>2004-07-15 11:20:00,574816</td>
<td>192.70.106.144</td>
<td>192.70.106.151</td>
<td>LDAP</td>
<td>MsgId=20 Bind Result</td>
</tr>
</tbody>
</table>
Typical scenarios

- System join to an Active Directory domain
- Domain member or domain controller startup
- Machine account password change
  - Every 30 days by default
- User authentication on a domain
- Domain controllers replication
- Group Policy applications
- ...

DNS and CLDAP traffic

- DNS traffic
  - SRV records resolution
    - `_service._protocol.DnsDomainName`
    - Ex: `_ldap._tcp.sitename._sites.dc._msdcs.domainname` to locate a domain controller inside a given site

- CLDAP traffic
  - Obtaining the closest domain controller
    - `DsGetDcName()` API, implemented by a pseudo RPC call to Active Directory
      - Site name is kept in cache (`DynamicSiteName` registry value)
  - ethereal display filter: `ldap && udp`
  - Documented in the *Locating Active Directory Servers* section of Windows 2000 Resource Kit documentation
DNS traffic: dynamic updates (1/2)

- DNS dynamic updates
  - Implemented by the dhcp service (even if IP address is static)
    - Register this connection's addresses in DNS (enabled by default)
    - at machine startup with static IP address (A and PTR)
    - at each IP address change with dynamic IP address (DHCP)
      - Depends on DHCP server configuration (by default, only A record)
    - each 24 hours by default
      - DefaultRegistrationRefreshInterval registry value
      - Default TTL of 20 minutes for updated records {A, PTR}
        (DefaultRegistrationTtl registry value)
    - manual registration: ipconfig /registerdns
DNS traffic: dynamic updates (2/2)

Ethernet II, Src: 00:0c:29:1f:da:98, Dst: 00:10:dc:ca:f3:53
Internet Protocol, Src Addr: 192.70.106.146 (192.70.106.146), Dst Addr: 192.70.106.151 (192.70.106.151)
Transmission Control Protocol, Src Port: 1053 (1053), Dst Port: 53 (53), Seq: 1461, Ack: 1, Len: 1181

Domain Name System (query)
  - Length: 2639
  - Transaction ID: 0x7ae7
  - Flags: 0x0000 (Standard query)
  - Questions: 1
  - Answer RRs: 0
  - Authority RRs: 0
  - Additional RRs: 1

Queries

Additional records
- 972-ms-7.1-15435.139f420e-4a8e-11d8-9694-000c291fda98: type TKEY, class any
  - Name: 972-ms-7.1-15435.139f420e-4a8e-11d8-9694-000c291fda98
  - Type: Transaction Key
  - Class: any
  - Time to live: 0 time
  - Data length: 2503
  - Algorithm name: gss-tsig
  - Signature inception: Aug 30, 2004 16:08:33,0000000000
  - Signature expiration: Aug 31, 2004 16:08:33,0000000000
  - Mode: GSSAPI
  - Error: No error
  - Key Size: 2477

Key Data
- GSS-API
  - OID: 1.3.6.1.5.5.2 (SNMPv2-SMI::security.5.2) (SPNEGO - Simple Protected Negotiation)
  - SPNEGO
    - negTokenInit
      - mechType
      - mechToken
        - krb5_blob: 6082096706032a864882f71201020201...
        - OID: 1.2.840.113554.1.2.2 (iso.2.840.113554.1.2.2) (KRB5 - Kerberos 5)
        - krb5_tok_id: KRB5_AP_REQ (0x0001)
LDAP traffic

- LDAP traffic
  - typically authenticated using the GSS-SPNEGO SASL mechanism
  - empty dn (*distinguished name*) in LDAP bind
  - starts with a request to obtain certain attributes of the RootDSE
    - `SupportedSASLMechanisms`
    - `LdapServiceName`
  - LDAP traffic can be encrypted
  - Examination of search parameters when traffic is unencrypted
    - Base DN, scope, filters, attributes, ...
  - LDAP request errors
    - `ldap.result.errormsg` display filter
MSRPC traffic (1/2)

- MSRPC traffic
  - RPC services localization over TCP/IP
    - endpoint mapper, TCP port 135 (epm)
    - Returns the TCP port on which a given RPC service is listening
    - map operation, unauthenticated
  - Local Security Authority access (lsa)
    - Kerberos authentication
    - TCP port (typically 1025, must be set to a static port, as documented in MSKB #224196)
    - Ex: LsarQueryInformationPolicy(2) operations
  - Active Directory access, using SAM RPC interface (samr)
    - Kerberos authentication, using same TCP port as LSA access
    - Ex: machine account creation on a DC for a new member server is implemented using the SamrCreateUser2InDomain operation
MSRPC traffic (2/2)

- MSPRC traffic (cont.)
  - Authentication on the domain, using netlogon service (rpc_netlogon)
    - Same TCP port as LSA and SAM access
    - NetrServerReqChallenge and NetrServerAuthenticate3 operations
  - Active Directory access, using RPC (instead of LDAP)
    - drsuapi interface, using the same TCP port
    - DRSCrackNames operation (DrsBind and DrsUnbind), implementing the DsCrackNames() API
  - Encrypted traffic, currently not decoded by ethereal
Kerberos traffic

- Obtaining a TGT (Ticket Granting Ticket)
  - Startup of a domain member server
  - User authentication
  - AS-REQ (10) and AS-REP (11) messages
- Obtaining service tickets
  - TGS-REQ (12) and TGS-REP (13) messages
  - Typical service names: host, ldap, cifs, dns, ...
- Using service tickets
  - AP-REQ (14) and AP-REP (15) messages
  - Typically encapsulated inside SPNEGO
Active Directory Service Principal Names (SPN)

- Service Principal Names
  - Kerberos authentication to Active Directory network services is implemented requesting a ticket for a given service
  - A service is designated using a SPN (Service Principal Name)
  - `servicePrincipalName` attribute (case-insensitive) in the User Active Directory object class
  - Also, `sPNMappings` attribute (list of equivalent SPNs to the host SPN)

- On the wire
  - SPN appear in TGS-REQ, TGS-REP and AS-REQ messages
  - A TGS-REP message can contain a different SPN from the one requested
    - Canonicalization option in Windows 2000
    - Returned SPN is similar to SERVER$
  - Canonicalization is disabled in Windows Server 2003
Registered SPN on a AD DC

C:\>setspn -L SERVEUR
Registered ServicePrincipalNames for CN=SERVEUR,OU=Domain Controllers,DC=DomaineBlah,DC=com:
  NtFrS-88f5d2bd-b646-11d2-a6d3-00c04fc9b232/serveur.DomaineBlah.com
  DNS/serveur.DomaineBlah.com
  ldap/serveur.DomaineBlah.com/ForestDnsZones.DomaineBlah.com
  HOST/serveur.DomaineBlah.com/DOMAINEBLAH
  HOST/SERVEUR
  HOST/serveur.DomaineBlah.com
  HOST/serveur.DomaineBlah.com/DomaineBlah.com
  E3514235-4B06-11D1-AB04-00C04FC2DCD2/276d4866-4940-49e4-91ec-991746baf84a/DomaineBlah.com
  ldap/276d4866-4940-49e4-91ec-991746baf84a._msdcs.DomaineBlah.com
  ldap/serveur.DomaineBlah.com/DOMAINEBLAH
  ldap/SERVEUR
  ldap/serveur.DomaineBlah.com
Kerberos tickets of a domain user (Windows 2000)

C:\>klist tickets
Cached Tickets: <5>

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
  KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
  End Time: 8/27/2004 0:51:47
  Renew Time: 9/2/2004 14:51:47

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
  KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
  End Time: 8/27/2004 0:51:47
  Renew Time: 9/2/2004 14:51:47

Server: HOST/serveur.DomaineBlah.com@DOMAINEBLAH.COM
  KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
  End Time: 8/27/2004 0:51:47
  Renew Time: 9/2/2004 14:51:47

Server: ldap/serveur.DomaineBlah.com/DomaineBlah.com@DOMAINEBLAH.COM
  KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
  End Time: 8/27/2004 0:51:47
  Renew Time: 9/2/2004 14:51:47

Server: LDAP/serveur.DomaineBlah.com@DOMAINEBLAH.COM
  KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
  End Time: 8/27/2004 0:51:47
  Renew Time: 9/2/2004 14:51:47
Kerberos tickets of a domain user (Windows XP)

C:\>klist tickets

Cached Tickets: <6>

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27

Server: ldap/serveur.DomaineBlah.com/DomaineBlah.com@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27

Server: cifs/serveur.domaineblah.com@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27

Server: LDAP/serveur.DomaineBlah.com@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27

Server: host/wxpdfilt.domaineblah.com@DOMAINEBLAH.COM
KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
End Time: 7/30/2004 5:56:27
Renew Time: 8/5/2004 19:56:27
C:\> klist tickets

Cached Tickets: <9>

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: krbtgt/DOMAINEBLAH.COM@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: W2KKDC$@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: SERVEUR$@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC<NT>
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Kerberos tickets on a domain controller (1/2) (LOCALSYSTEM logon session)
Kerberos tickets on a domain controller (2/2) (LOCALSYSTEM logon session)

Server: ldap/w2kdcc.DomaineBlah.com/DomaineBlah.com@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: HOST/serveur.DomaineBlah.com@DOMAINEBLAH.COM
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

Server: E3514279-4B06-11D1-AB04-00C04FC2D83D@276d4866-4940-49e4-91ec-9917546bafa/Domai
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    End Time: 9/7/2004 20:32:06
    Renew Time: 9/14/2004 10:32:06

C:\>
Kerberos traffic: errors

- Kerberos traffic: common errors
  - KRB-ERROR (30) messages (kerberos.msg.type == 30)
    - KRB5KRB_AP_ERR_SKEW
      - Time synchronization problem
    - KRB5KDC_ERR_PREAMUTH_FAILED
      - Pre-authentication error (typically, incorrect password)
    - KRB5KRB_AP_ERR_TKT_EXPIRED
      - Expired ticket, to be renewed
      - LSA keeps user passwords in cache and can request a new TGT, within a maximum limit of 7 days (Max. Lifetime for user ticket renewal)
    - KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN
      - Principal not recognized by the KDC
      - Missing SPN (servicePrincipalName attribute) in an AD account?
      - Also when an IP address is used in a UNC path
        - NTLM authentication fallback
Kerberos tickets decryption

Client Realm: WUNHINEBHM.COM

Ticket

Tkt-vno: 5
Realm: DOMAINEBHM.COM

Server Name (Service and Instance): LDAP/serveur.DomaaineBlah.com
- Name-type: Service and Instance (2)
- Name: LDAP
- Name: serveur.DomaaineBlah.com

enc-part rc4-hmac
- Encryption type: rc4-hmac (23)
- Kvec: 21
- enc-part: A9ADC3F96D13D09C26E76303D902B8F...

[Decrypted using: keytab principal serveur3@DOMAINEBHM.COM]

EncTicketPart

Padding: 0

Ticket Flags (Forwardable, Renewable, Pre-Auth, Ok As Delegate)

key rc4-hmac

Client Realm: DOMAINEBHM.COM

Client Name (Principal): vtsoin$

TransitedEncoding: DOMAIN-Y500-COMPRESS

Auth-time: 2004-08-27 13:32:15 (Z)
Start time: 2004-08-27 13:32:15 (Z)
End time: 2004-08-27 23:32:15 (Z)
Renew-till: 2004-09-03 13:32:15 (Z)

AuthorizationData AD-IF-RELEVENT

Type: AD-IF-RELEVENT (1)

Data: 308026250020025E900400200009182...

IF_RELEVENT AD-WinK-PAC

Type: AD-WinK-PAC (128)

Data: 040000000000000010000000000000...

- Num Entries: 4
- Version: 0
- Type: Logon Info (1)
- Type: Client Info Type (10)
- Type: Server Checksum (6)
- Type: Privil Checksum (7)

enc-part rc4-hmac

Encryption type: rc4-hmac (23)

enc-part: 71C9A30038E43139DB4A4F32FD1DA7...

[Decrypted using: key learnt from frame 79]
Active Directory replication

× Active Directory replication

× `drsuapi` MSRPC interface (one TCP port)

× Restricting Active Directory Replication Traffic to a Specific Port (MSKB #224196)

× Between domain controllers

× `DRSReplicaSync` operation (`drsuapi`)
  × Used to notify a replication partner that updates are available for replication

× `DRSGetNCChanges` operation (`drsuapi`)
  × Used to obtain updates for a given AD Naming Context

× RPC connection to the `drsuapi` service are authenticated using a Kerberos ticket obtained for the following principal:

× `e3514235-4b06-11d1-ab04-00c04fc2dcd2` (`drsuapi` interface UUID)

× Destination domain controller GUID

× DNS domain name
FRS replication (File Replication Service)

- FRS replication
  - frsrpc MSRPC interface (1 TCP port)
  - How to Restrict FRS Replication Traffic to a Specific Static Port (MSKB #319553)
- Between domain controllers
  - FrsRpcStartPromotionParent operation at DC startup
  - FrsRpcSendCommPkt operation for updates replication
NTP traffic

- NTP traffic
  - w32time service, started on domain member servers
  - NT5DS mode (by default), using AD hierarchy for time synchronization
  - NTP synchronization at startup, with a domain controller
    - Identified using CLDAP at system startup
    - Each 45 minutes (3 times), then each 8 hours
  - Synchronization mechanism
    - Client sends the RID of the machine account in the request, using the KeyID field
      - This RID was previously obtained in the response of the NetrServerAuthenticate3 operation
    - Timestamp is signed (message authentication code field)
Other approaches

• Limitations of the network analysis approach
  • With encrypted traffic: LDAP and certain MSRPC operations
  • Traffic not properly dissected by the network analyzer
    • Typically with MSRPC, where RPC operations do not contain enough information to identify the DCE RPC interface
      \[ \text{→ ethereal Decode As DCE-RPC function} \]

• Other approaches
  • Correlation of network traces and logged events
    • Security and System event log of Windows systems
  • Diagnostic tools on servers
    • Ex: NTDS object statistics using the System Monitor tool (perfmon.msc)
    • Ex: tools to examine Kerberos tickets cache
Conclusion

- A good understanding of aforementioned protocols is needed to deploy Active Directory
- Network analysis is one of the possible way to obtain this understanding
  - Looking at these protocols on the wire, in a real environment, is a good complement to technical whitepapers reading
- Network analysis can also be used to diagnose anomalies
  - When diagnostic tools or logfiles are not enough...
- ethereal is a tool of choice to analyse network traces obtained in Active Directory environments
References: network traffic

- Network traffic in Windows environments
  - Windows 2000 Startup and Logon Traffic Analysis
  - Network Ports Used by Key Microsoft Server Products
  - Using Windows { XP SP1, 2000 SP4, Server 2003} in a Managed Environment
    - http://go.microsoft.com/fwlink/?LinkId={22607, 22608, 22609}
References: DNS

* DNS implementation in Active Directory
  * Windows 2000 DNS White Paper
  * RFC 3645: Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)
References: Kerberos

- Protocol
  - draft-ietf-krb-wg-kerberos-clarifications-08.txt
  - RFC 1510 update (original specification of Kerberos V)
  - http://kerberos.info/

- Documents
  - Troubleshooting Kerberos Errors (Microsoft)

- Tools
  - klist, kerbtray (Microsoft)
  - tktview: http://msdn.microsoft.com/msdnmag/issues/0500/security/
  - leash32: http://web.mit.edu/kerberos/
References: LDAP

* LDAP and CLDAP
  * Active Directory Domain Controller Location Service (Anthony Liguori, Samba team)
    * CLDAP description (Connectionless LDAP)
  * Active Directory LDAP compliance (Microsoft)
    * http://www.microsoft.com/windowsserver2003/techinfo/overview/ldapcomp.mspx
  * Active Directory LDAP schema (Windows 2000, Windows Server 2003 and ADAM)
References: SMB/CIFS and MSRPC

- Reference book on SMB/CIFS
  - Implementing CIFS
    - http://www.ubiqx.org/cifs/
- MSRPC
  - Windows network services internals
    - http://www.hsc.fr/ressources/articles/win_net_srv/
  - Testing MSRPC (Andrew Tridgell, Samba Team)
  - MSRPC architecture & security problems related
  - Microsoft Windows RPC Security Vulnerabilities
    - http://conference.hackinthebox.org/materials/lsd/
References: SNTP

x Microsoft references

x The Windows Time Service
  x http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/operate/wintime.mspx

x Basic Operation of the Windows Time Service (MSKB #224799)

x Windows Time Service Tools and Settings (Windows Server 2003 Technical Reference)

x Using Windows XP Professional with Service Pack 1 in a Managed Environment (Windows Time Service)
  x http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/xpmanaged/27_xpwts.mspx

x Security aspects of time synchronization infrastructure
  x http://www.security.nnov.ru/advisories/timesync.asp
Greetings

- Emmanuel Le Chevoir and Fabien Dupont
- ethereal developpers community