NAC@ACK

Michael Thumann

&

Dror-John Roecher

Agenda

- **Part 1 – Introduction (very short)**
  - Some marketing buzz on Cisco NAC
- **Part 2 – NAC Technology**
  - All you need to know about NAC (in order to hack it)
- **Part 3 – Security Analysis**
  - Delving into the security flaws of Ciscos‘ NAC solution
- **Part 4 – Approaching NAC@ACK**
  - The stony road towards a working exploit
- **Part 5 - Showtime**

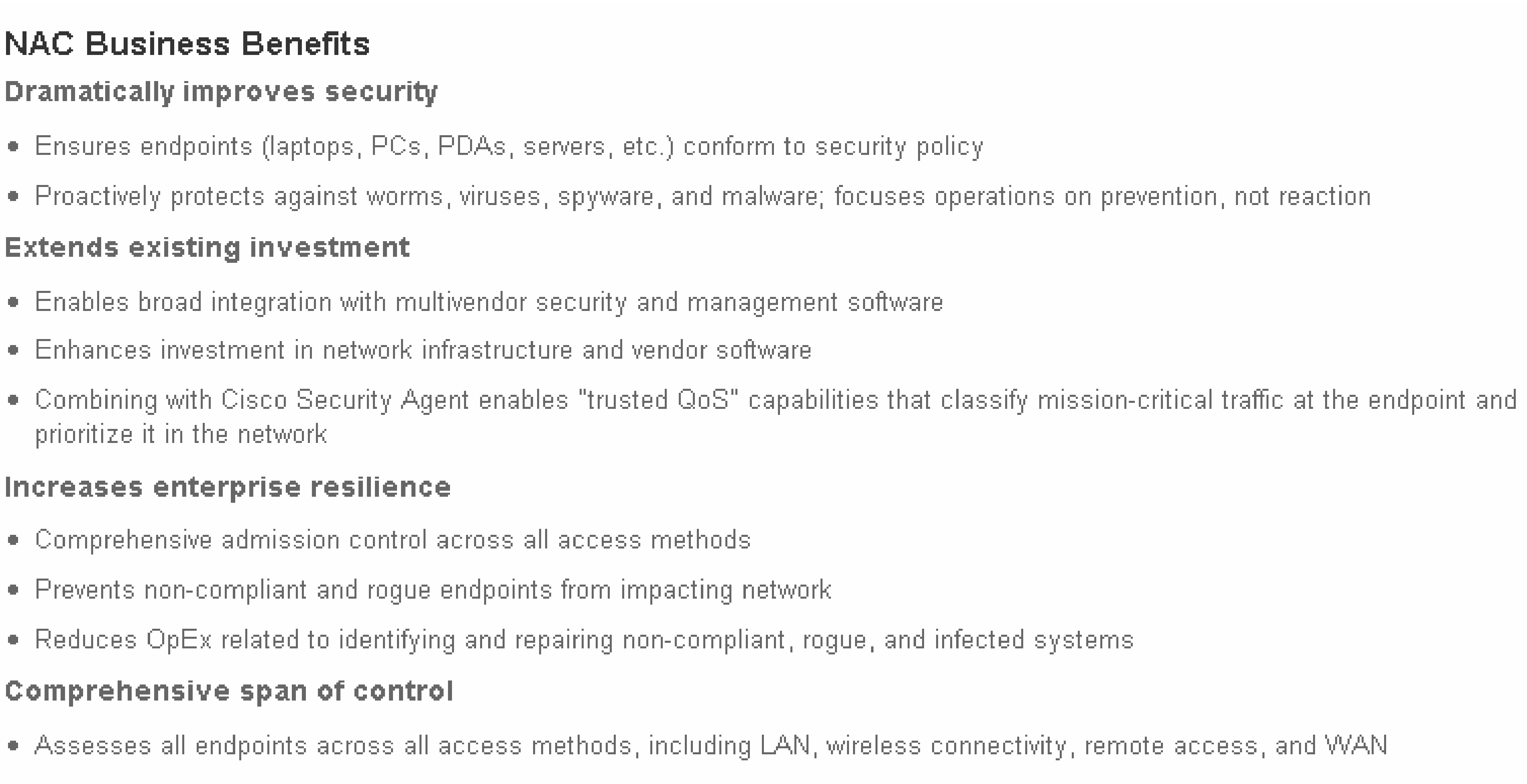
ERNW
Wir leben IT-Security.

Part 1 - Introduction

Why is Cisco selling Cisco NAC?

- **Because customers are willing to pay for it ,-)**
- **But why are customers willing to pay for it?**
- **Because Cisco makes some pretty cool promises… see next slide**

From: http://www.cisco.com/go/nac



NAC Business Benefits

Dramatically improves security

- Ensures endpoints (laptops, PCs, PDAs, servers, etc.) conform to security policy
- Proactively protects against worms, viruses, spyware, and malware; focuses operations on prevention, not reaction

Extends existing investment

- Enables broad integration with multivendor security and management software
- Enhances investment in network infrastructure and vendor software
- Combining with Cisco Security Agent enables "trusted QoS" capabilities that classify mission-critical traffic at the endpoint and prioritize it in the network

Increases enterprise resilience

- Comprehensive admission control across all access methods
- Prevents non-compliant and rogue endpoints from impacting network
- Reduces OpEx related to identifying and repairing non-compliant, rogue, and infected systems

Comprehensive span of control

- Assesses all endpoints across all access methods, including LAN, wireless connectivity, remote access, and WAN

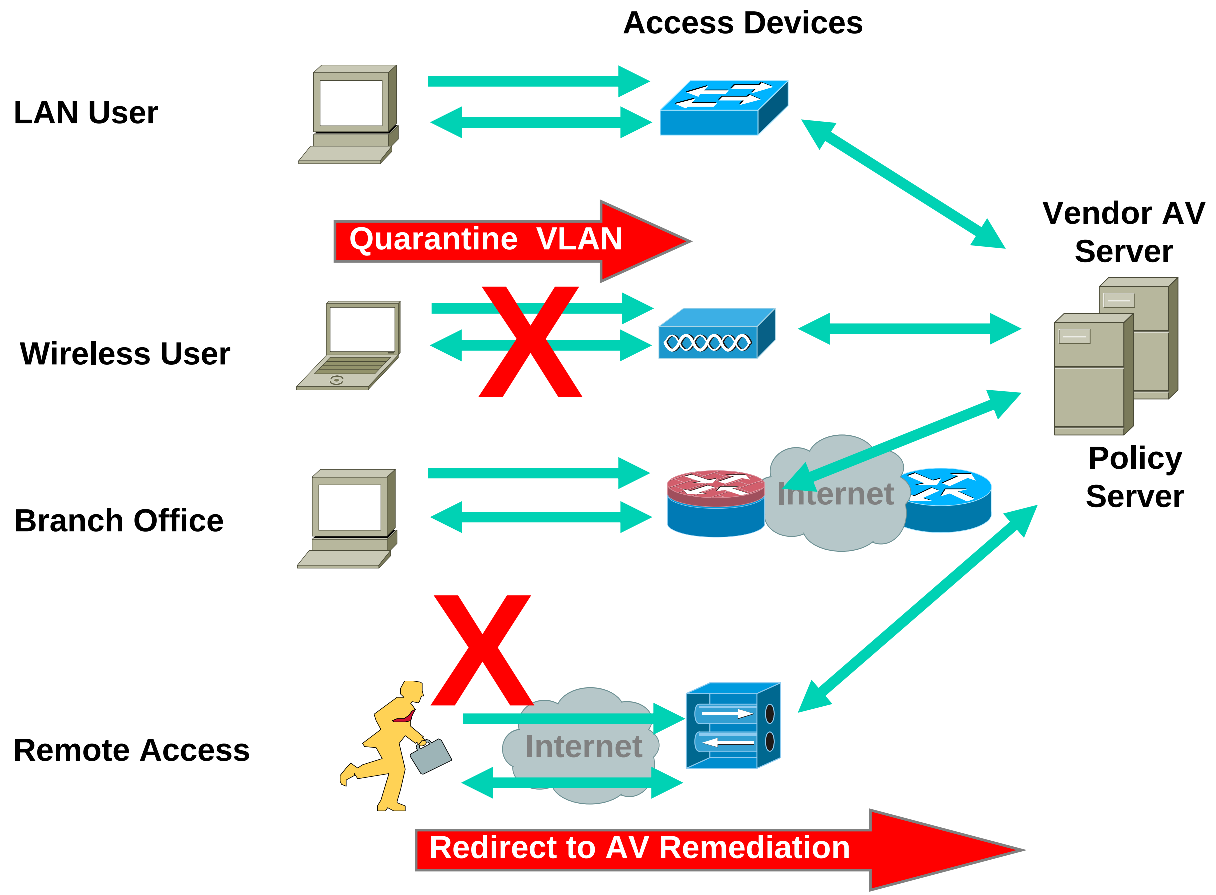
The idea behind Cisco NAC

- **Grant access to the network based on the grade of compliance to a defined (security) policy. So it is first of all a compliance solution and not a security solution.**

- **Security Policy can usually be broken down to:**
  - Patch level (OS & Application)
  - AV signatures & scan engine up to date
  - No „unwanted“ programs (e.g. l33t t00ls)
  - Desktop Firewall up & running

- **If a client is non-compliant to the policy [and is not whitelisted somewhere – think network-printers], restrict access.**

Policy based Access…

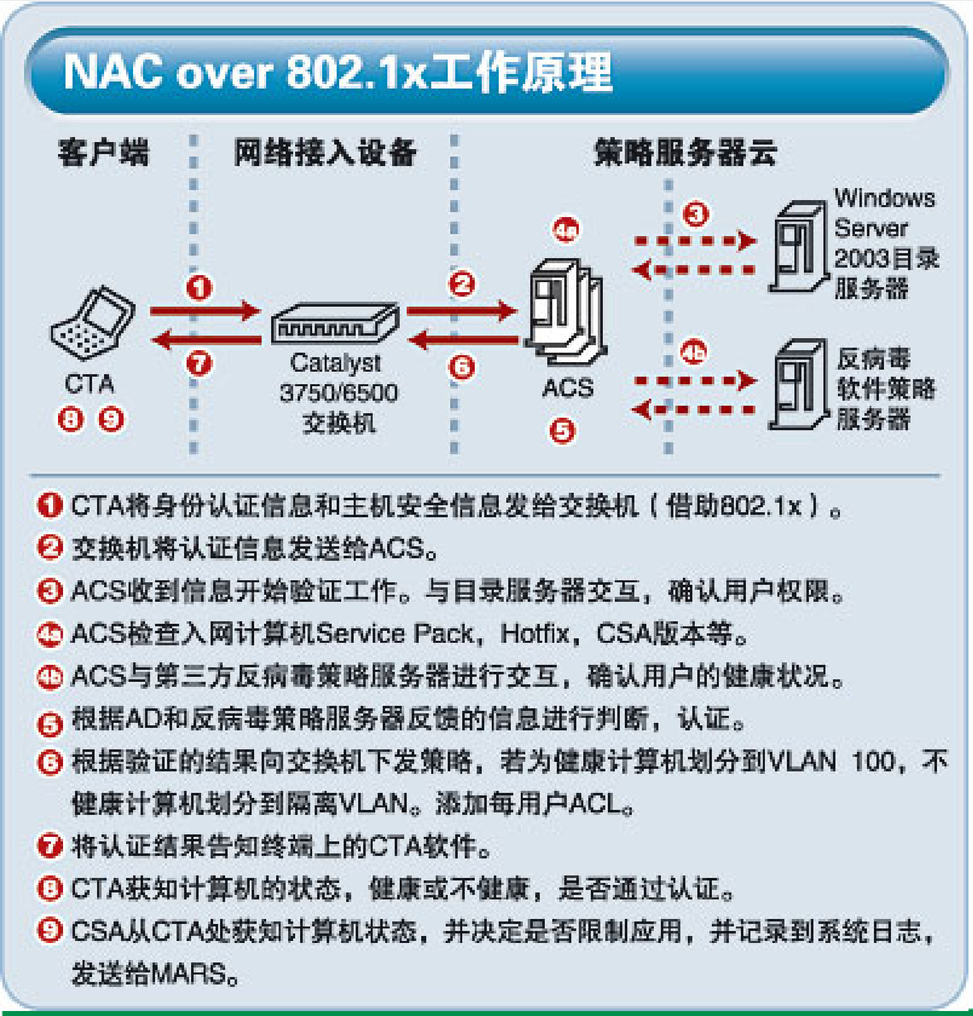


1. Access Device detects new client.
2. Access Device queries the client for an agent and relays information to a backend policy server.
3. Policy Server checks received information against defined rules and derives an appropriate access-level
4. Access-Device enforces restrictions

Wir leben IT-Security.

Part 2 – NAC Technology

What is Cisco NAC?



NAC over 802.1x工作原理

❶ CTA将身份认证信息和主机安全信息发给交换机（借助802.1x）。

❷ 交换机将认证信息发送给ACS。

❸ ACS收到信息开始验证工作。与目录服务器交互，确认用户权限。

4a ACS检查入网计算机Service Pack，Hotfix，CSA版本等。

4b ACS与第三方反病毒策略服务器进行交互，确认用户的健康状况。

❺ 根据AD和反病毒策略服务器反馈的信息进行判断，认证。

❻ 根据验证的结果向交换机下发策略，若为健康计算机划分到VLAN 100，不健康计算机划分到隔离VLAN。添加每用户ACL。

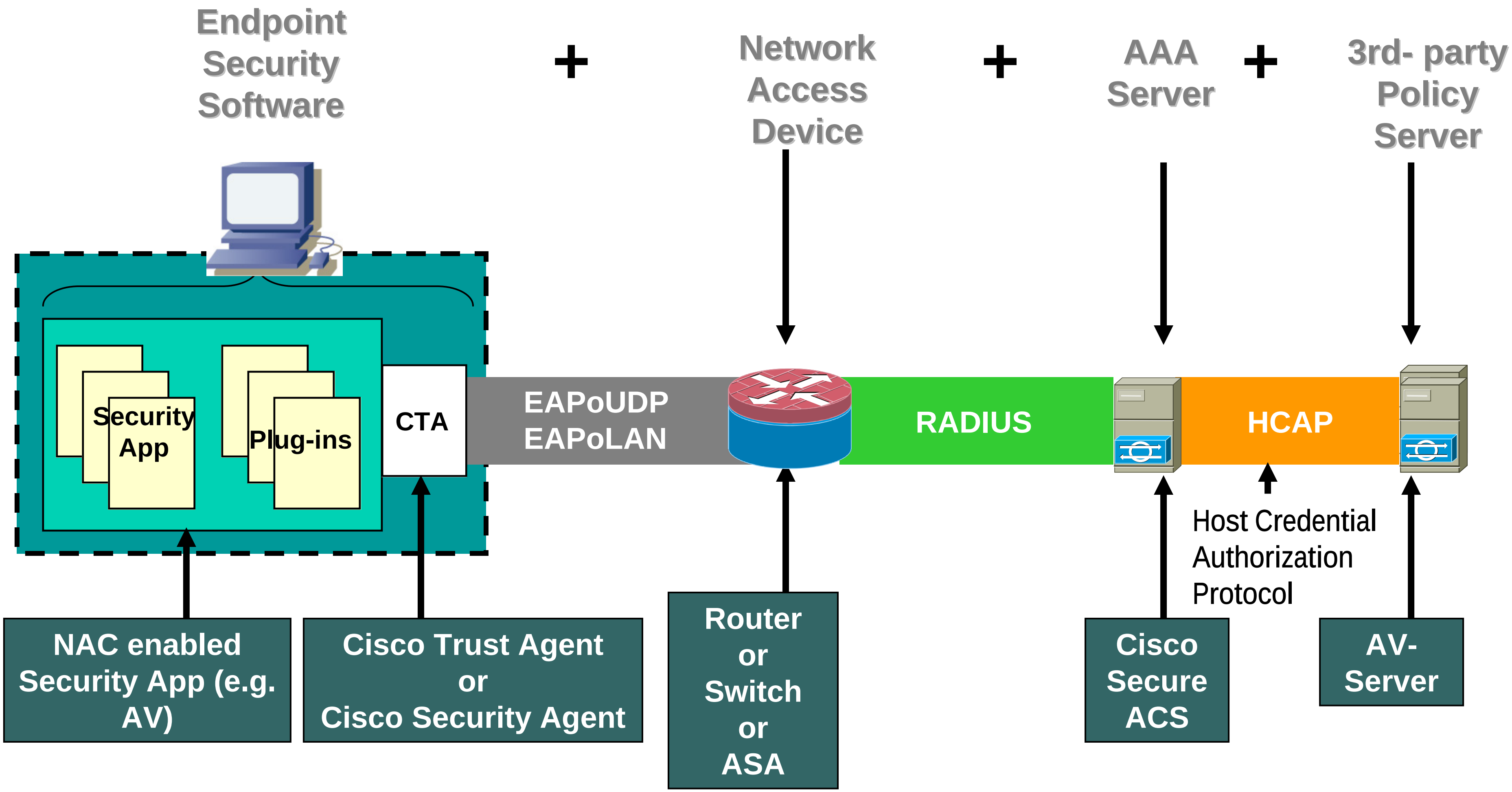
❼ 将认证结果告知终端上的CTA软件。

❽ CTA获知计算机的状态，健康或不健康，是否通过认证。

❾ CSA从CTA处获知计算机状态，并决定是否限制应用，并记录到系统日志，发送给MARS。

?

A „big overview“ picture…



There are 3 different NAC flavours…

- **NAC-Layer3-IP**
  - Access-restrictions are implemented as IP-ACLs
  - NAD is a Layer-3 device (e.g. a Router or a VPN-Concentrator/Firewall).
  - The communication takes place using PEAP over EAP over UDP (EoU).
- **NAC-Layer2-IP**
  - Access-restrictions as IP-ACLs on a VLAN-interface of a switch.
  - The communication takes place using PEAP over EAP over UDP (EoU)
- **NAC-Layer2-802.1x**
  - Uses 802.1x port control to restrict network access
  - Obviously the device enforcing these restrictions is a switch.
  - EAP-FAST is used in conjunction with 802.1x.
  - This is the only NAC flavour where the client is:
    - authenticated before being allowed on the network
    - restricted from communicating with its local subnet

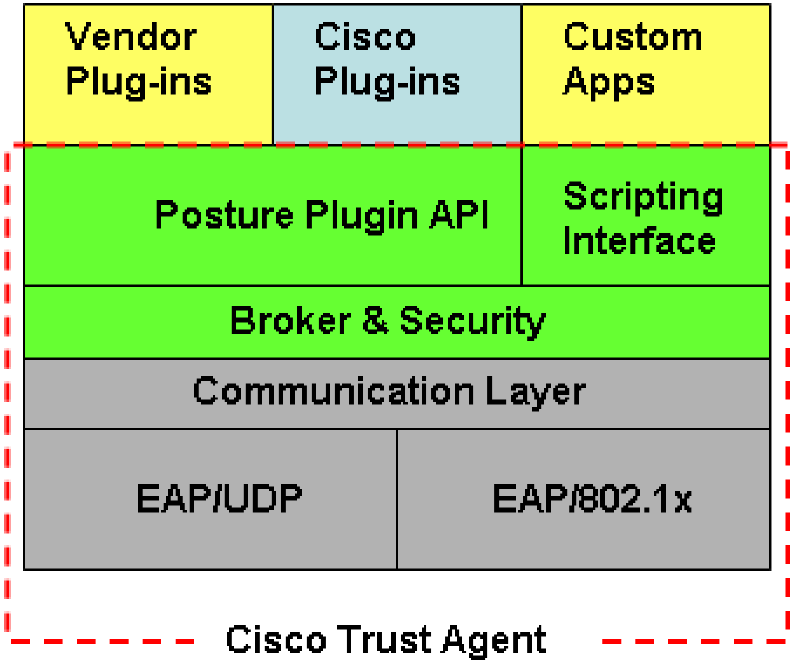
(Some) Features…

Feature	NAC-L2-802.1x	NAC-L2-IP	NAC-L3-IP
Trigger	Data Link / Switchport	DHCP / ARP	Routed Packet
Machine ID	Yes	No	No
User ID	Yes	No	No
Posture	Yes	Yes	Yes
VLAN Assignment	Yes	No	No
URL Redirection	No	Yes	Yes
Downloadable ACLs	Cat65k only	Yes	Yes

Yet another agent: Cisco Trust Agent

- **The Cisco Trust Agent (CTA) is the main component of the NAC framework installed on the clients.**
- **Its‘ tasks are to collect „posture data“ about the client and forward it to the ACS via the NAD.**
- **It has a plug-in interface for 3rd party vendors‘ NAC-enabled applications.**
- **It has a scripting interface for self-written scripts.**

CTA architecture



Vendor Plug-ins

Cisco Plug-ins

Custom Apps

Posture Plugin API

Scripting Interface

Broker & Security

Communication Layer

EAP/UDP

EAP/802.1x

Cisco Trust Agent

- **The CTA comes with two plug-ins by default:**
  - Cisco:PA
  - Cisco:Host

Posture Information

- **The information collected are Attribute-Value-pairs categorized by**
  - Vendor: ID based on IANA SMI assignement
  - Application-Type: see next slide
  - Credential Name: e.g. “OS Version”
  - Value-Format: String, Date, etc.
- **For all plug-ins & scripts this information is collected in a plaintext “.inf-file”.**

Application Types in Cisco NAC

Application-Type ID	Application-Type Name	Usage
1	PA	Posture Agent
2	Host / OS	Host information
3	AV	Anti Virus
4	FW	Firewall
5	HIPS	Host IPS
6	Audit	Audit
32768 – 65536		Reserved for “local use” (custom plug-ins or scripts)

Credentials for Cisco:PA & Cisco:Hosts

Application-Type	Attribute Number	Attribute Name	Value-Type
Posture Agent	3 4 5 6 7 8 9	Agent-Name (PA-Name) Agent-Version OS-Type OS-Version User-Notification OS-Kernel OS-Kernel-Version	String Version String Version String String Version
Host	11	Machine-Posture-State	1 – Booting, 2 – Running, 3 – Logged in.
	6 7 8	Service Packs Hot Fixes Host-FQDN	String String String

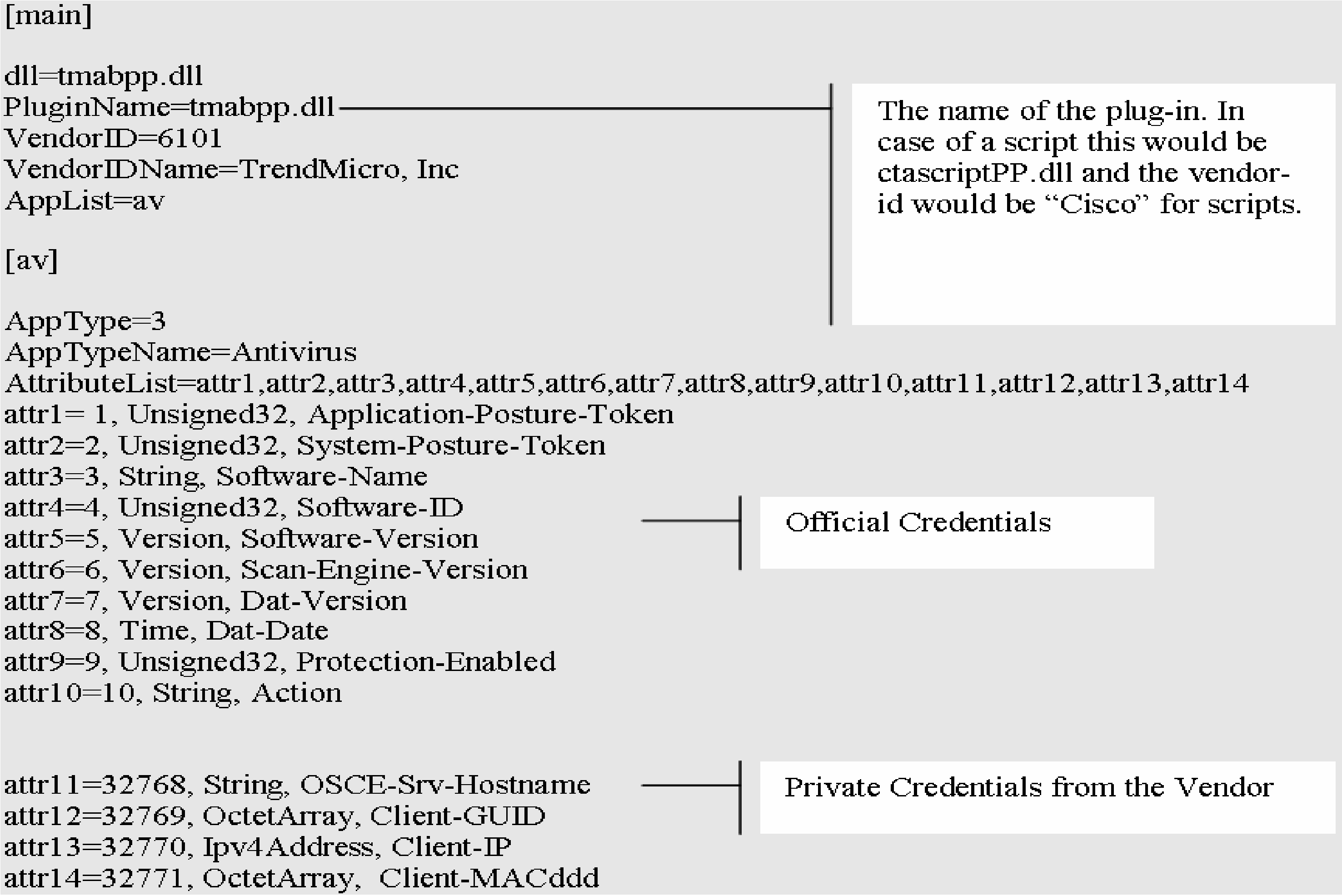
Posture Tokens…

- **For each plug-in/Application/script an “Application Posture Token” (APT) is derived by the ACS through the configured policy.**
- **This token is one out of:**
  - Healthy, Checkup, Quarantine, Transition, Infected, Unknown (see next slide for definitions of these tokens)
- **From all APTs a “System Posture Token” (SPT) is derived – this corresponds to the APT which will grant the least access on the network to the client.**
- **The SPT is associated with access-restrictions on the ACS (e.g. downloadable ACL, URL-Redirection).**

Posture Tokens – well defined

- “Healthy”: fully compliant with the admission policy for the specified application.
- “Checkup”: partial but sufficient compliance with the admission policy, no need to restrict access, a warning to the user may be issued.
- “Transition”: either during boot-time, when not all necessary services have been started or during an audit-process for clientless hosts, temporary access-restrictions may be applied.
- “Quarantine”: insufficient compliance with the admission policy, network access is usually restricted to a quarantine/remediation segment.
- “Infected”: active infection detected, usually most restrictive network access even up to complete isolation.
- “Unknown”: a token can not be determined or no CTA installed on client. This may lead to partial access (guest-vlan & internet-access for example).

Sample inf-File for Trendmicro AV



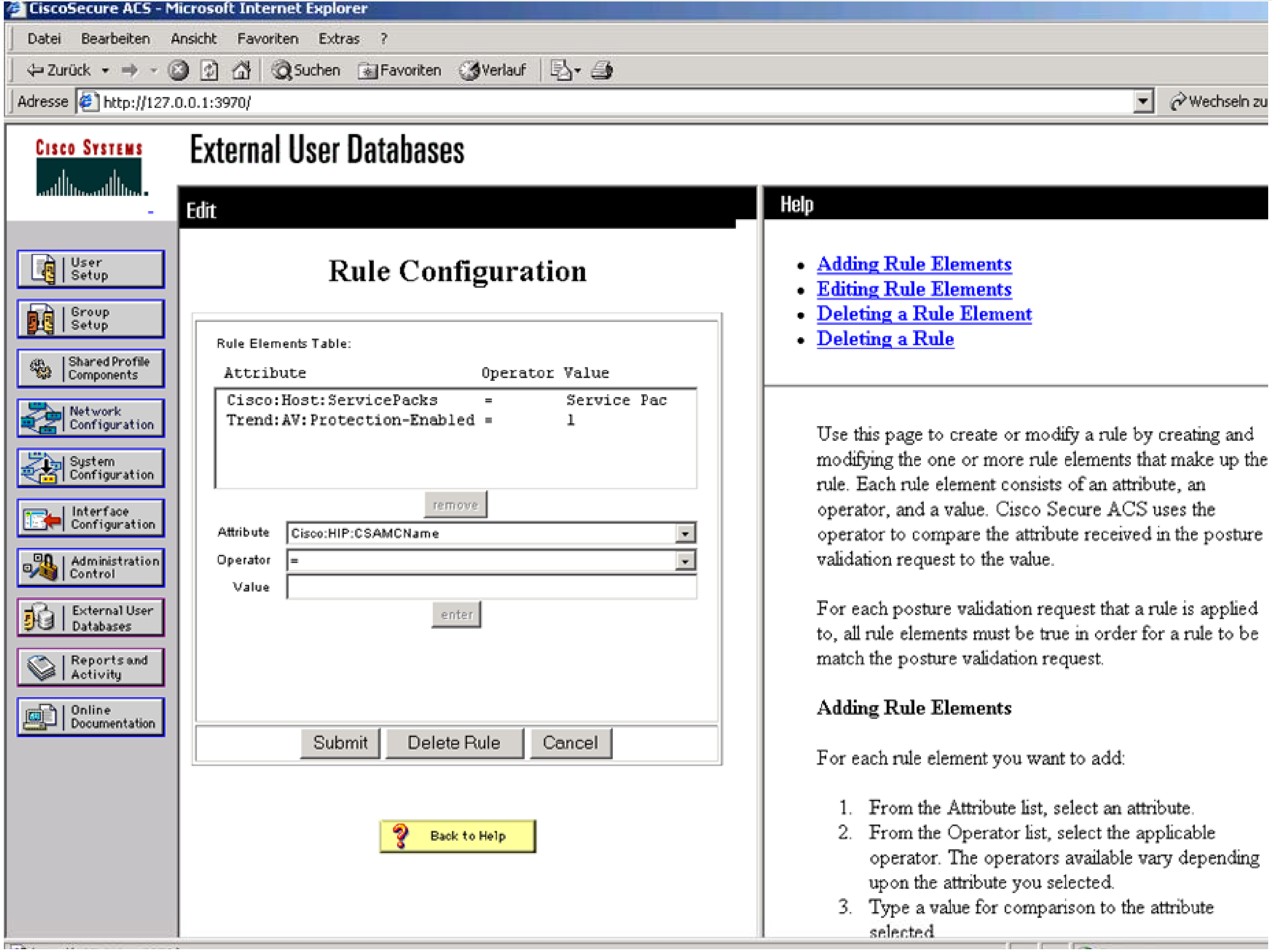
```
[main]

dll=tmabpp.dll
PluginName=tmabpp.dll
VendorID=6101
VendorIDName=TrendMicro, Inc
AppList=av

[av]

AppType=3
AppTypeName=Antivirus
AttributeList=attr1,attr2,attr3,attr4,attr5,attr6,attr7,attr8,attr9,attr10,attr11,attr12,attr13,attr14
attr1= 1, Unsigned32, Application-Posture-Token
attr2=2, Unsigned32, System-Posture-Token
attr3=3, String, Software-Name
attr4=4, Unsigned32, Software-ID
attr5=5, Version, Software-Version
attr6=6, Version, Scan-Engine-Version
attr7=7, Version, Dat-Version
attr8=8, Time, Dat-Date
attr9=9, Unsigned32, Protection-Enabled
attr10=10, String, Action


attr11=32768, String, OSCE-Srv-Hostname
attr12=32769, OctetArray, Client-GUID
attr13=32770, Ipv4Address, Client-IP
attr14=32771, OctetArray,  Client-MACddd
```

The name of the plug-in. In case of a script this would be ctascriptPP.dll and the vendor-id would be “Cisco” for scripts.

Official Credentials

Private Credentials from the Vendor

Sample Policy on Cisco ACS



External User Databases

Rule Configuration

Rule Elements Table:

```
Attribute                        Operator Value
Cisco:Host:ServicePacks       =          Service Pac
Trend:AV:Protection-Enabled =          1
```

Attribute: Cisco:HIP:CSAMCName

Operator: =

Value:

- Adding Rule Elements
- Editing Rule Elements
- Deleting a Rule Element
- Deleting a Rule

Use this page to create or modify a rule by creating and modifying the one or more rule elements that make up the rule. Each rule element consists of an attribute, an operator, and a value. Cisco Secure ACS uses the operator to compare the attribute received in the posture validation request to the value.

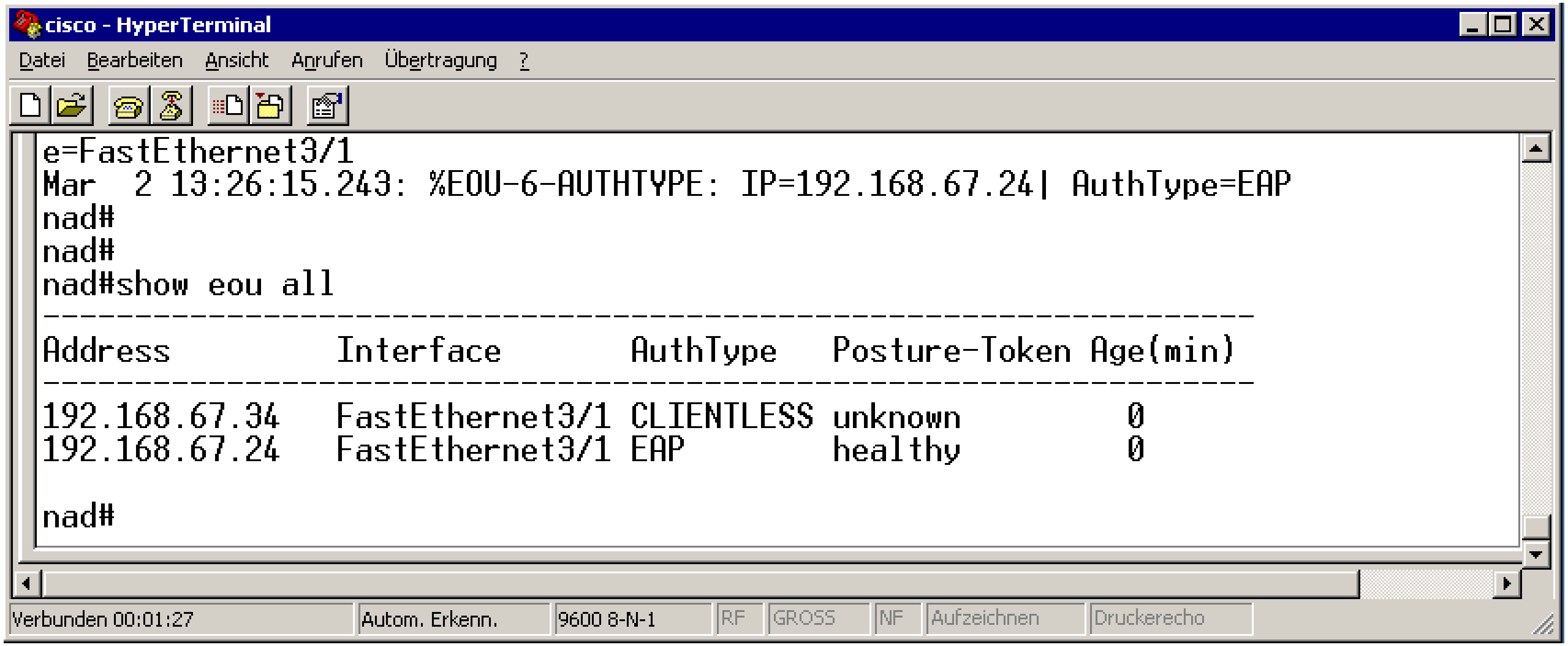
For each posture validation request that a rule is applied to, all rule elements must be true in order for a rule to be match the posture validation request.

Adding Rule Elements

For each rule element you want to add:

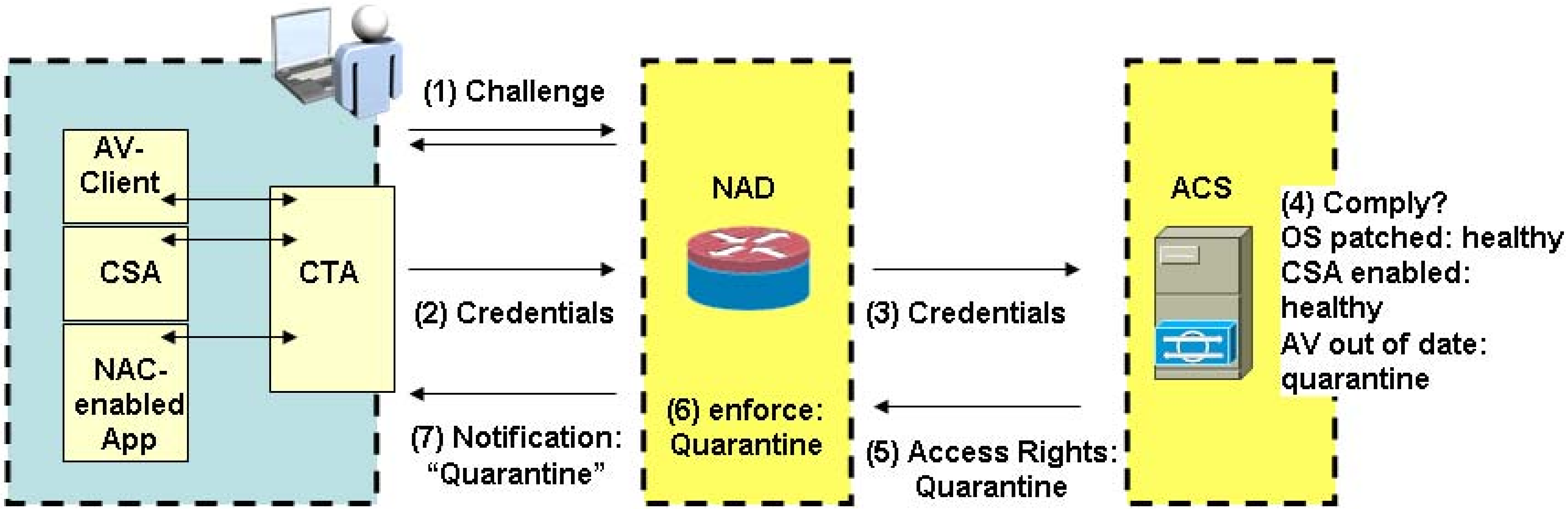
1. From the Attribute list, select an attribute.
2. From the Operator list, select the applicable operator. The operators available vary depending upon the attribute you selected.
3. Type a value for comparison to the attribute selected.

And the resulting SPT on a NAD



```
e=FastEthernet3/1
Mar  2 13:26:15.243: %EOU-6-AUTHTYPE: IP=192.168.67.24| AuthType=EAP
nad#
nad#
nad#show eou all
-------------------------------------------------------------------
Address          Interface        AuthType   Posture-Token Age(min)
-------------------------------------------------------------------
192.168.67.34    FastEthernet3/1 CLIENTLESS unknown          0
192.168.67.24    FastEthernet3/1 EAP        healthy          0

nad#
```

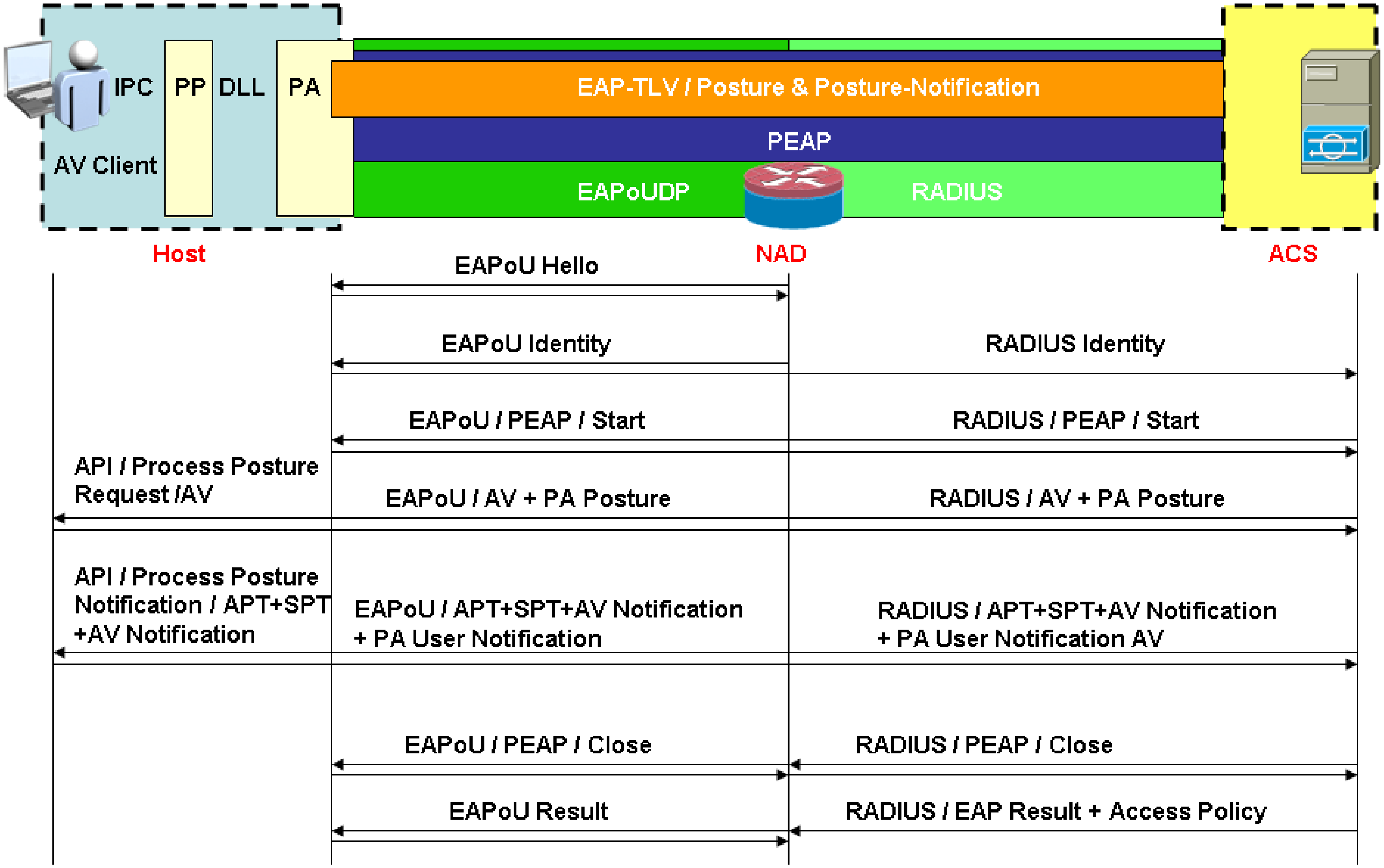
General Communication Flow



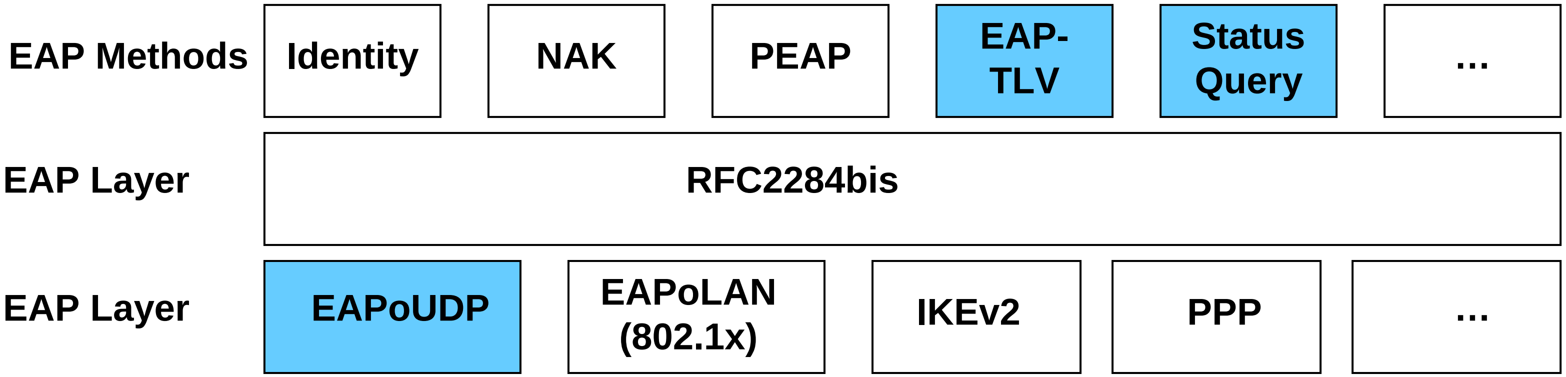
Transport Mechanisms…

- **NAC-Layer2-802.1x**
  - Uses 802.1x
  - Uses EAP-FAST as EAP method
  - Uses EAP-TLV to transport posture information
- **NAC-Layer2-IP**
  - Uses EAP over UDP (Port 21862 on client & NAD)
  - Uses PEAPv1 as EAP method without inner authentication
  - Uses EAP-TLV to transport posture information
- **NAC-Layer3-IP**
  - Uses EAP over UDP (Port 21862 on client & NAD)
  - Uses PEAPv1 as EAP method without inner authentication
  - Uses EAP-TLV to transport posture information

NAC-L3-IP Communication Flow


IPC
PP
DLL
PA
AV Client
EAP-TLV / Posture & Posture-Notification
PEAP
EAPoUDP
RADIUS
Host
NAD
ACS
EAPoU Hello
EAPoU Identity
RADIUS Identity
EAPoU / PEAP / Start
RADIUS / PEAP / Start
API / Process Posture Request /AV
EAPoU / AV + PA Posture
RADIUS / AV + PA Posture
API / Process Posture Notification / APT+SPT +AV Notification
EAPoU / APT+SPT+AV Notification + PA User Notification
RADIUS / APT+SPT+AV Notification + PA User Notification AV
EAPoU / PEAP / Close
RADIUS / PEAP / Close
EAPoU Result
RADIUS / EAP Result + Access Policy

Extensible Authentication Protocol

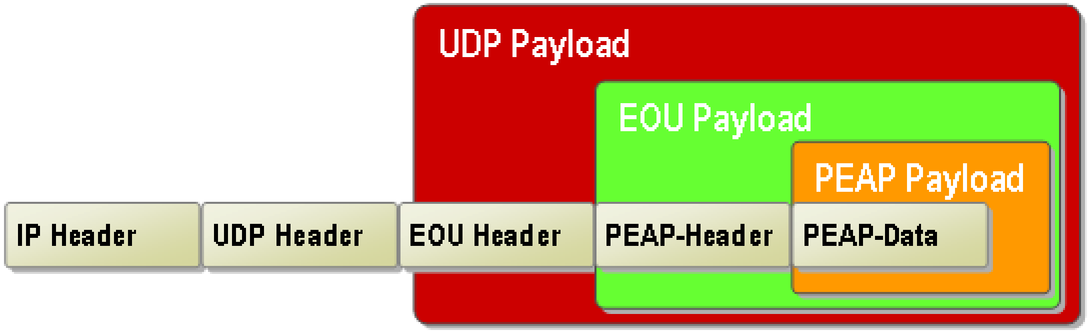


EAP Methods	Identity	NAK	PEAP	EAP-TLV	Status Query	…
EAP Layer	RFC2284bis					
EAP Layer	EAPoUDP	EAPoLAN (802.1x)	IKEv2	PPP	…	

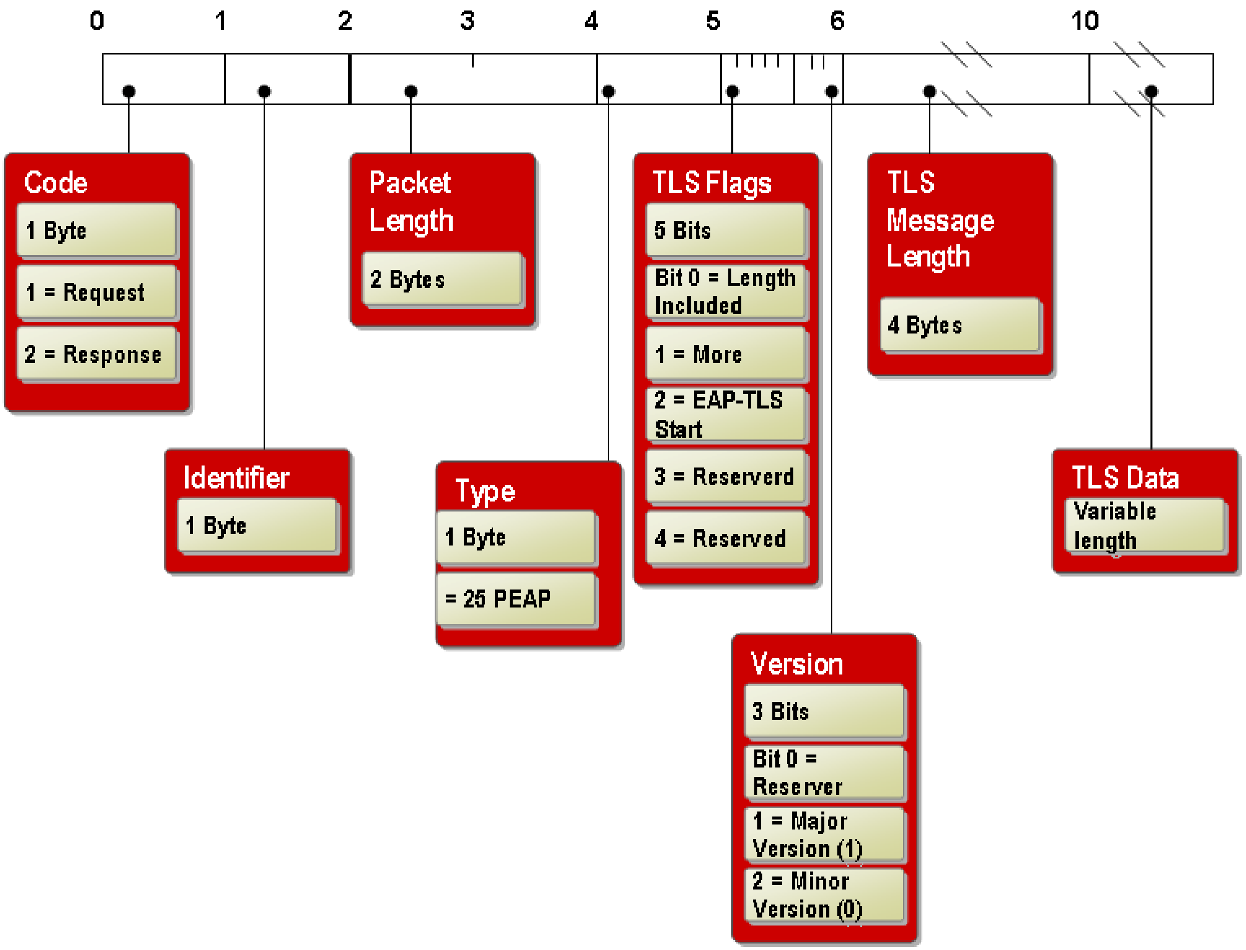
New Function

- EAP is a“request-response” Protocol:
  - **Exchange of “identity” and “authentication” information between a supplicant and an AAA server.**
- EAP supports a multitude auf authentication-schemes
  - **EAP-MD5**
  - **EAP-MSCHAP**
  - **…**
- EAP has to be “enhanced” for “policy based access restrictions” (aka NAC)
  - ***EAP-TLV*: Attribute-Type-Length-Value-Pair**
  - ***Status Query*: new method to get query the state of a client**
  - ***EAPoUDP*: EAP Transport over IP (instead of over Layer2 as e.g. 802.1x)**

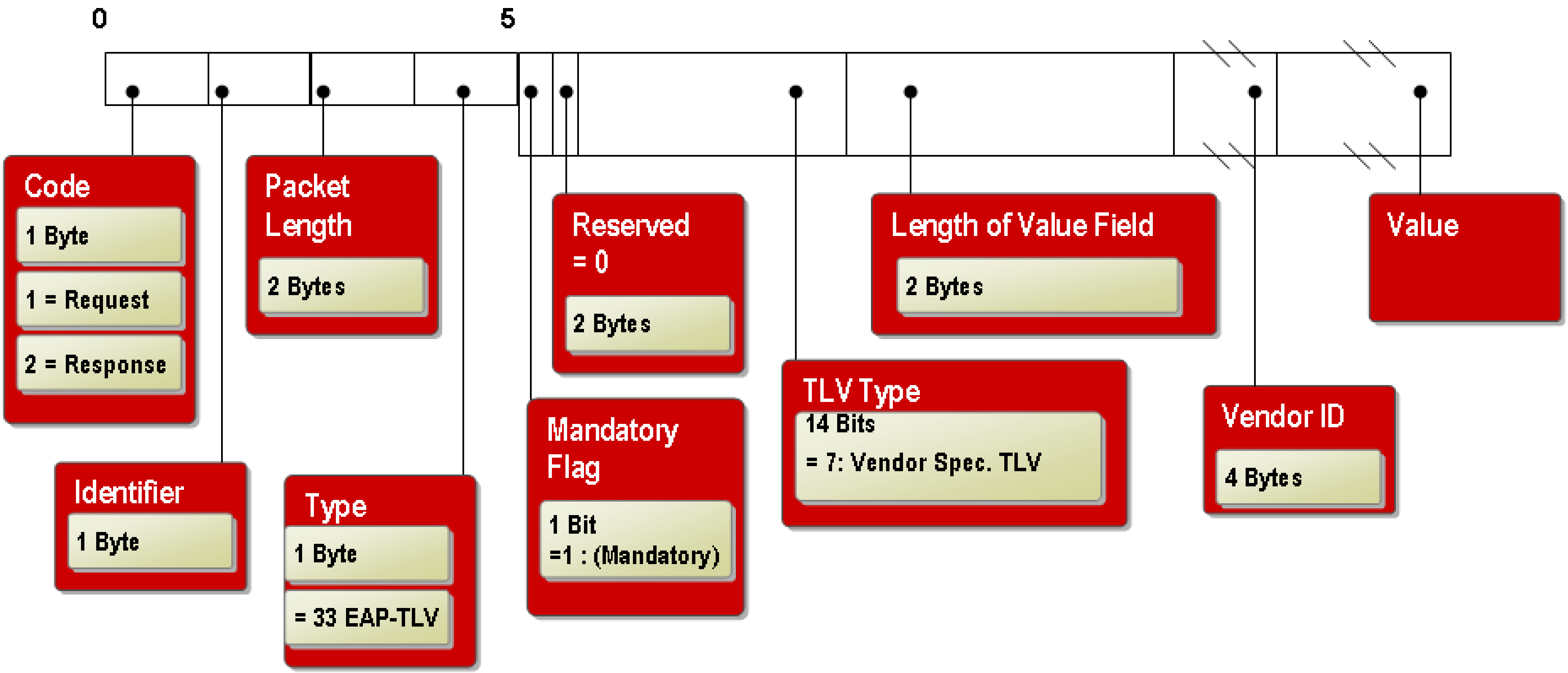
Encapsulation for L2-IP & L3-IP



PEAPv1 Frame Format



EAP-TLV Vendor Frame Format



Part 3 – Security Analysis

Flawed by Design 1:Client Authentication

ERNW
Wir leben IT-Security.

	NAC-Layer 3 IP	NAC Layer 2 IP	NAC Layer 2 802.1x
Client Authentication	**No intrinsic Client Authentication**. In VPN scenarios there is a "VPN Authentication" which might be considered a "mitigating control".	**No intrinsic Client Authentication** – and no means of "adding" such on top.	Client Authentication based on 802.1x/EAP-FAST
Restriction of access on local subnet.	It is not possible to restrict access to the local subnet via NAC.	It is not possible to restrict access to the local subnet via NAC.	Access to local subnet can be denied through "port shutdown" via NAC.

Flawed by Design

- **So 1st design flaw is :**

Authorization without Authentication

- **This is clearly breaking a “secure by design” approach [for a security product] and is not conforming to “Best Current Practices”**

Flawed by Design 2: Epimenides Paradox

- **Epimenides was a Cretan (philosopher) who made one statement: "All Cretans are liars."**

- **Same paradox applies to Cisco NAC as well:**
  - The goal is to judge the “compliance”-level of (un)known & untrusted clients.
  - This is achieved by asking the (un)known & untrusted client about itself.
  - How can the ACS be sure that the client is a Cretan philosopher (a liar)?

So what? Where is the attack?

Posture Spoofing Attack

- **We define "posture spoofing" as an attack where a legitimate or illegitimate client spoofs "NAC posture credentials" in order to get unrestricted network access.**

Attackers Definition - Insider

- **Insider: An insider is a legitimate user of a NAC-protected network. The client has a working installation of the CTA and valid user/machine-credentials for the network. Additionally the inside attacker has the certificate of the ACS installed in its certificate store and if 802.1x is being used, this attacker has valid EAP-FAST-Credentials (PAC).**
- **The insider simply wants to bypass restrictions placed on his machine (e.g. no “leet tools” allowed and NAC checks list of installed programs).**

Attackers Definition - Outsider

- **Outsider: An outsider is not a legitimate user of the NAC-protected network and wants to get unrestricted access to the network. The outsider has no valid user/machine-credentials and no working CTA installation.**

Attack Vectors

- **Code an “alternative” NAC client**
  - Definitly possible
  - Will not work on 802.1x with EAP-FAST for outsider.
  - Currently “development in process” ☺
- **Replace plug-ins with self-written ones**
  - Definitely possible (be patient for ~50 more slides *just kidding*)
  - Works for the “insider” but not for the “outsider”.
  - Less work than the “alternative client
- **Abuse the scripting interface**
  - Not verified yet – limitations on “Vendor-ID” and “Application-ID” apply and not (yet) known if these are enforced or can be circumvented
  - If possible – the easiest way ☺

Feasible Attack Vectors

	Insider	**Outsider**
NAC-L2-802.1x	DLL/Plug-In replacement Scripting Interface CTA replacement	None as to our current knowledge.
NAC-L2-IP	DLL/Plug-In replacement Scripting Interface CTA replacement	CTA replacement
NACL-L3-IP	DLL/Plug-In replacement Scripting Interface CTA replacement	CTA replacement

Part 4 – Approaching NAC@AK

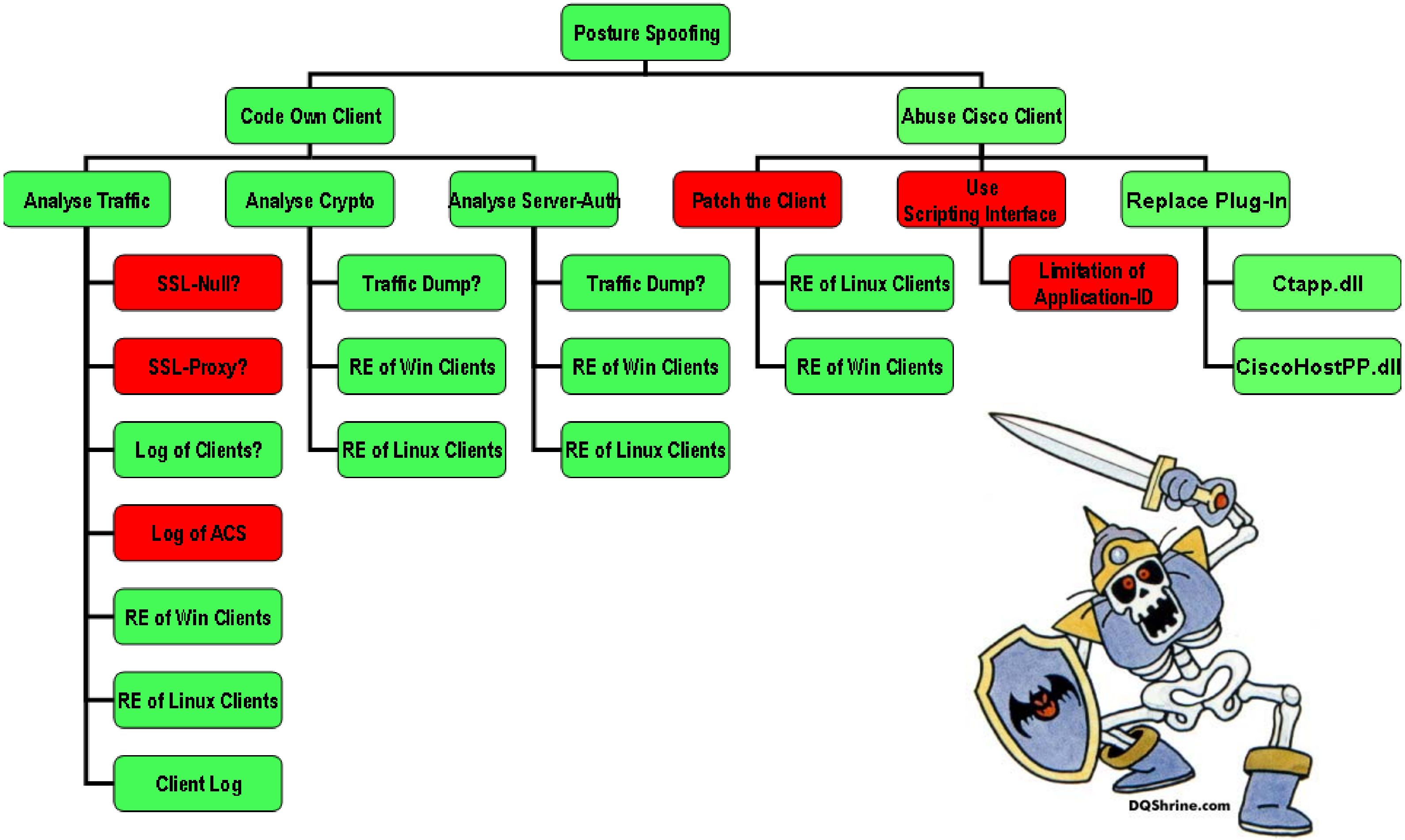
The ugly stuff – working with a structured approach *sigh

- **Step 1: Define what you need to know in order to get it working.**
- **Step 2: Sketch an attack-tree showing steps towards the goal.**
- **Step 3: Evaluate the components of the attack-tree for feasibility. Get the “tools” & know the “techniques” you need.**
- **Step 4: Pursue the feasible steps from step 3.**
- **Step 5: loop to step (1) until you get it working ,-)**

Want to know

- **Everything relating to…**
  - Communication flow
  - Packet format
  - Data-structures
  - Used Crypto
  - Used libraries
  - Existing interfaces
  - Program flow
  - Used Authentication
  - …

Attack Tree



- Posture Spoofing
  - Code Own Client
    - Analyse Traffic
      - SSL-Null?
      - SSL-Proxy?
      - Log of Clients?
      - Log of ACS
      - RE of Win Clients
      - RE of Linux Clients
      - Client Log
    - Analyse Crypto
      - Traffic Dump?
      - RE of Win Clients
      - RE of Linux Clients
    - Analyse Server-Auth
      - Traffic Dump?
      - RE of Win Clients
      - RE of Linux Clients
  - Abuse Cisco Client
    - Patch the Client
      - RE of Linux Clients
      - RE of Win Clients
    - Use Scripting Interface
      - Limitation of Application-ID
    - Replace Plug-In
      - Ctapp.dll
      - CiscoHostPP.dll

Tools & Techniques

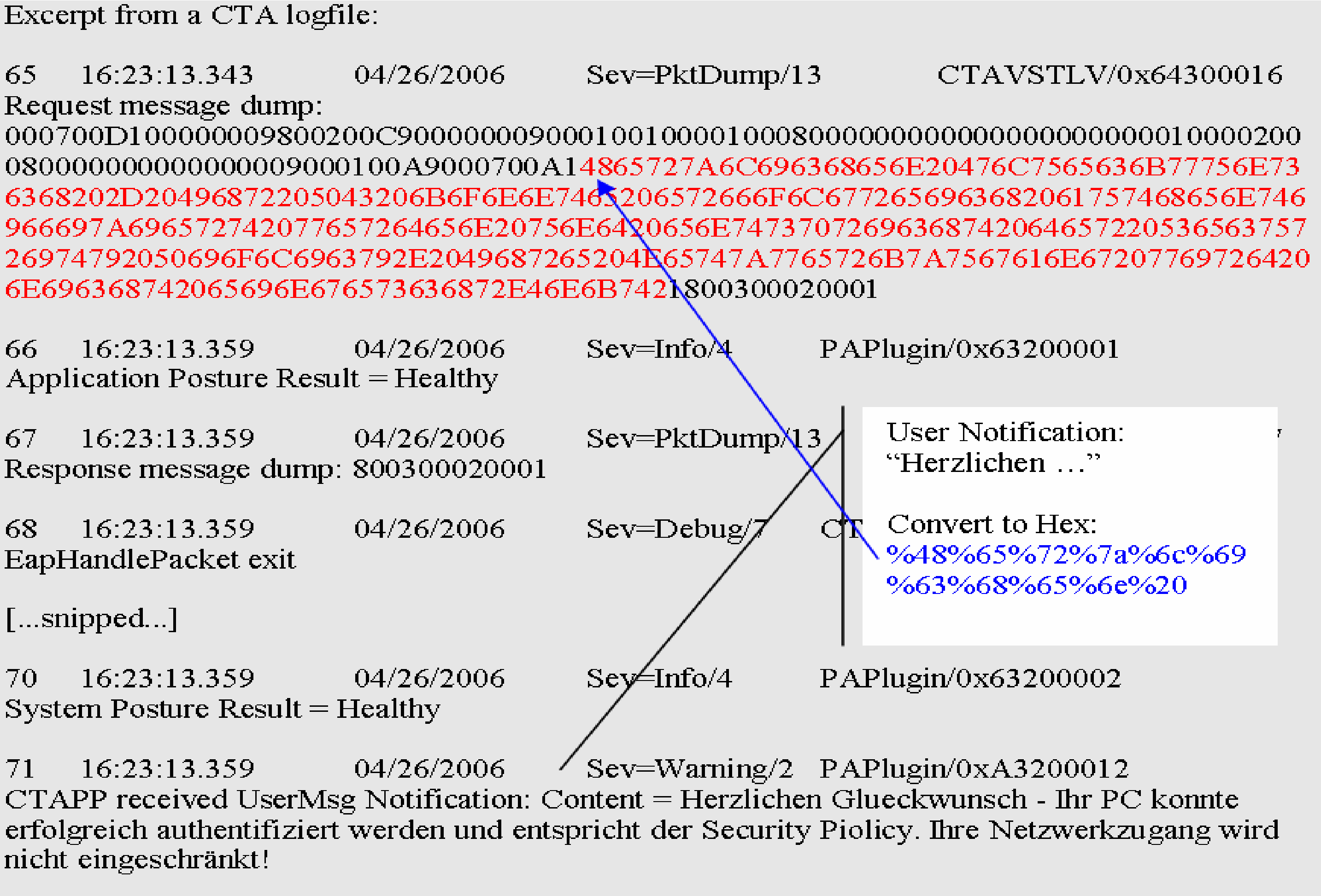
- **Reverse Engineering**
  - Reverse Engineering aims at uncovering the constructional elements of a product. IDAPro ☺ … and Hex-Rays
- **Packet Sniffing**
  - You all know that - Wireshark/Ethereal
- **Packet Diffing**
  - Extracting common and differing parts of two packets.
- **Debugging / API-Monitoring / Function-Hooking**
  - Through attaching a debugger or api-monitor to the running process, it is possible to actually see the contents of the stack while the program is running.
- **Built-in capabilities**
  - Logging / Debugging capabilites of the product – Cisco is usually _very_ good at that!
- **RTFM**
  - Read Read Read – often then vendor will tell you a lot about the product.

Big “want to have”: Cleartext Packets…

- **Communication is encrypted using TLS… packet capture shows encrypted packets.**
- **Not possible to get cleartext dump with tools (SSLProxy, etc.) – TLS over UDP not supported by tools.**

- **RTFM: Client Log can be enabled and it can dump cleartext payload of packets *g**

Cleartext Packet Dump in Log



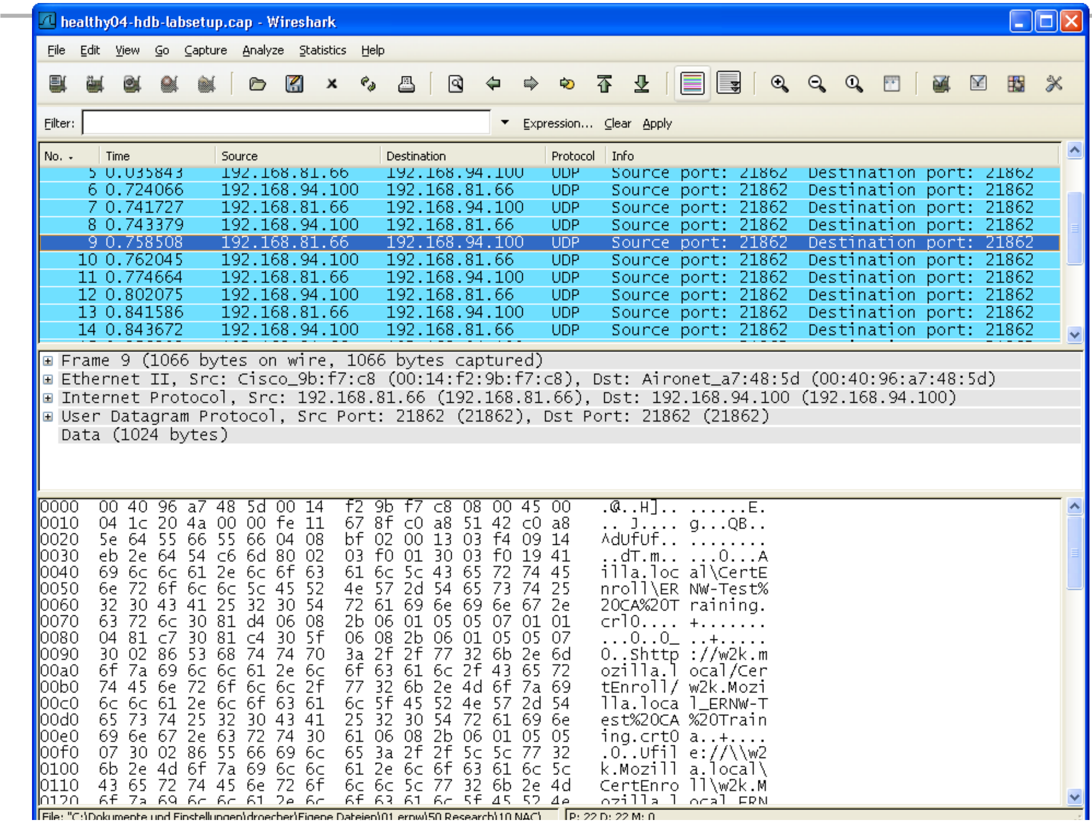
Excerpt from a CTA logfile:

```
65     16:23:13.343          04/26/2006        Sev=PktDump/13          CTAVSTLV/0x64300016
Request message dump:
000700D1000000009800200C900000000900010010000100080000000000000000000000010000200
08000000000000000009000100A9000700A14865727A6C696368656E20476C7565636B77756E73
6368202D20496872205043206B6F6E6E7465206572666F6C6772656963682061757468656E746
966697A696572742077657264656E20756E6420656E747370726963687420646572205365637 57
26974792050696F6C6963792E20496872652 04E65747A7765726B7A7567616E67207769726420
6E696368742065696E676573636872E46E6B7421800300020001

66     16:23:13.359          04/26/2006        Sev=Info/4         PAPlugin/0x63200001
Application Posture Result = Healthy

67     16:23:13.359          04/26/2006        Sev=PktDump/13
Response message dump: 800300020001

68     16:23:13.359          04/26/2006        Sev=Debug/7        CT

EapHandlePacket exit

[...snipped...]

70     16:23:13.359          04/26/2006        Sev=Info/4         PAPlugin/0x63200002
System Posture Result = Healthy

71     16:23:13.359          04/26/2006        Sev=Warning/2   PAPlugin/0xA3200012
CTAPP received UserMsg Notification: Content = Herzlichen Glueckwunsch - Ihr PC konnte
erfolgreich authentifiziert werden und entspricht der Security Piolicy. Ihre Netzwerkzugang wird
nicht eingeschränkt!
```

User Notification:
"Herzlichen …"

Convert to Hex:
%48%65%72%7a%6c%69
%63%68%65%6e%20

Packet Sniffing & Diffing

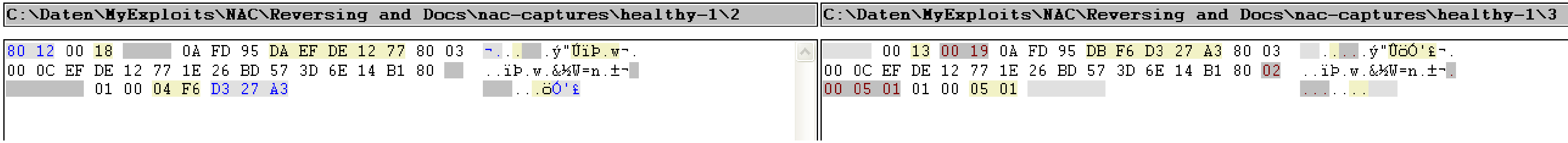

healthy04-hdb-labsetup.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time      Source          Destination     Protocol Info
  5 0.035843     192.168.81.66   192.168.94.100  UDP      Source port: 21862  Destination port: 21862
  6 0.724066     192.168.94.100  192.168.81.66   UDP      Source port: 21862  Destination port: 21862
  7 0.741727     192.168.81.66   192.168.94.100  UDP      Source port: 21862  Destination port: 21862
  8 0.743379     192.168.94.100  192.168.81.66   UDP      Source port: 21862  Destination port: 21862
  9 0.758508     192.168.81.66   192.168.94.100  UDP      Source port: 21862  Destination port: 21862
 10 0.762045     192.168.94.100  192.168.81.66   UDP      Source port: 21862  Destination port: 21862
 11 0.774664     192.168.81.66   192.168.94.100  UDP      Source port: 21862  Destination port: 21862
 12 0.802075     192.168.94.100  192.168.81.66   UDP      Source port: 21862  Destination port: 21862
 13 0.841586     192.168.81.66   192.168.94.100  UDP      Source port: 21862  Destination port: 21862
 14 0.843672     192.168.94.100  192.168.81.66   UDP      Source port: 21862  Destination port: 21862
```

```
⊞ Frame 9 (1066 bytes on wire, 1066 bytes captured)
⊞ Ethernet II, Src: Cisco_9b:f7:c8 (00:14:f2:9b:f7:c8), Dst: Aironet_a7:48:5d (00:40:96:a7:48:5d)
⊞ Internet Protocol, Src: 192.168.81.66 (192.168.81.66), Dst: 192.168.94.100 (192.168.94.100)
⊞ User Datagram Protocol, Src Port: 21862 (21862), Dst Port: 21862 (21862)
  Data (1024 bytes)
```

```
0000  00 40 96 a7 48 5d 00 14  f2 9b f7 c8 08 00 45 00   .@..H].. ......E.
0010  04 1c 20 4a 00 00 fe 11  67 8f c0 a8 51 42 c0 a8   .. J.... g...QB..
0020  5e 64 55 66 55 66 04 08  bf 02 00 13 03 f4 09 14   ^dUfUf.. ........
0030  eb 2e 64 54 c6 6d 80 02  03 f0 01 30 03 f0 19 41   ..dT.m.. ...0...A
0040  69 6c 6c 61 2e 6c 6f 63  61 6c 5c 43 65 72 74 45   illa.loc al\CertE
0050  6e 72 6f 6c 6c 5c 45 52  4e 57 2d 54 65 73 74 25   nroll\ER NW-Test%
0060  32 30 43 41 25 32 30 54  72 61 69 6e 69 6e 67 2e   20CA%20T raining.
0070  63 72 6c 30 81 d4 06 08  2b 06 01 05 05 07 01 01   crl0.... +.......
0080  04 81 c7 30 81 c4 30 5f  06 08 2b 06 01 05 05 07   ...0..0_ ..+.....
0090  30 02 86 53 68 74 74 70  3a 2f 2f 77 32 6b 2e 6d   0..Shttp ://w2k.m
00a0  6f 7a 69 6c 6c 61 2e 6c  6f 63 61 6c 2f 43 65 72   ozilla.l ocal/Cer
00b0  74 45 6e 72 6f 6c 6c 2f  77 32 6b 2e 4d 6f 7a 69   tEnroll/ w2k.Mozi
00c0  6c 6c 61 2e 6c 6f 63 61  6c 5f 45 52 4e 57 2d 54   lla.loca l_ERNW-T
00d0  65 73 74 25 32 30 43 41  25 32 30 54 72 61 69 6e   est%20CA %20Train
00e0  69 6e 67 2e 63 72 74 30  61 06 08 2b 06 01 05 05   ing.crt0 a..+....
00f0  07 30 02 86 55 66 69 6c  65 3a 2f 2f 5c 5c 77 32   .0..Ufil e://\\w2
0100  6b 2e 4d 6f 7a 69 6c 6c  61 2e 6c 6f 63 61 6c 5c   k.Mozill a.local\
0110  43 65 72 74 45 6e 72 6f  6c 6c 5c 77 32 6b 2e 4d   CertEnro ll\w2k.M
0120  6f 7a 69 6c 6c 61 2e 6c  6f 63 61 6c 5f 45 52 4e   ozilla.l ocal_ERN
```

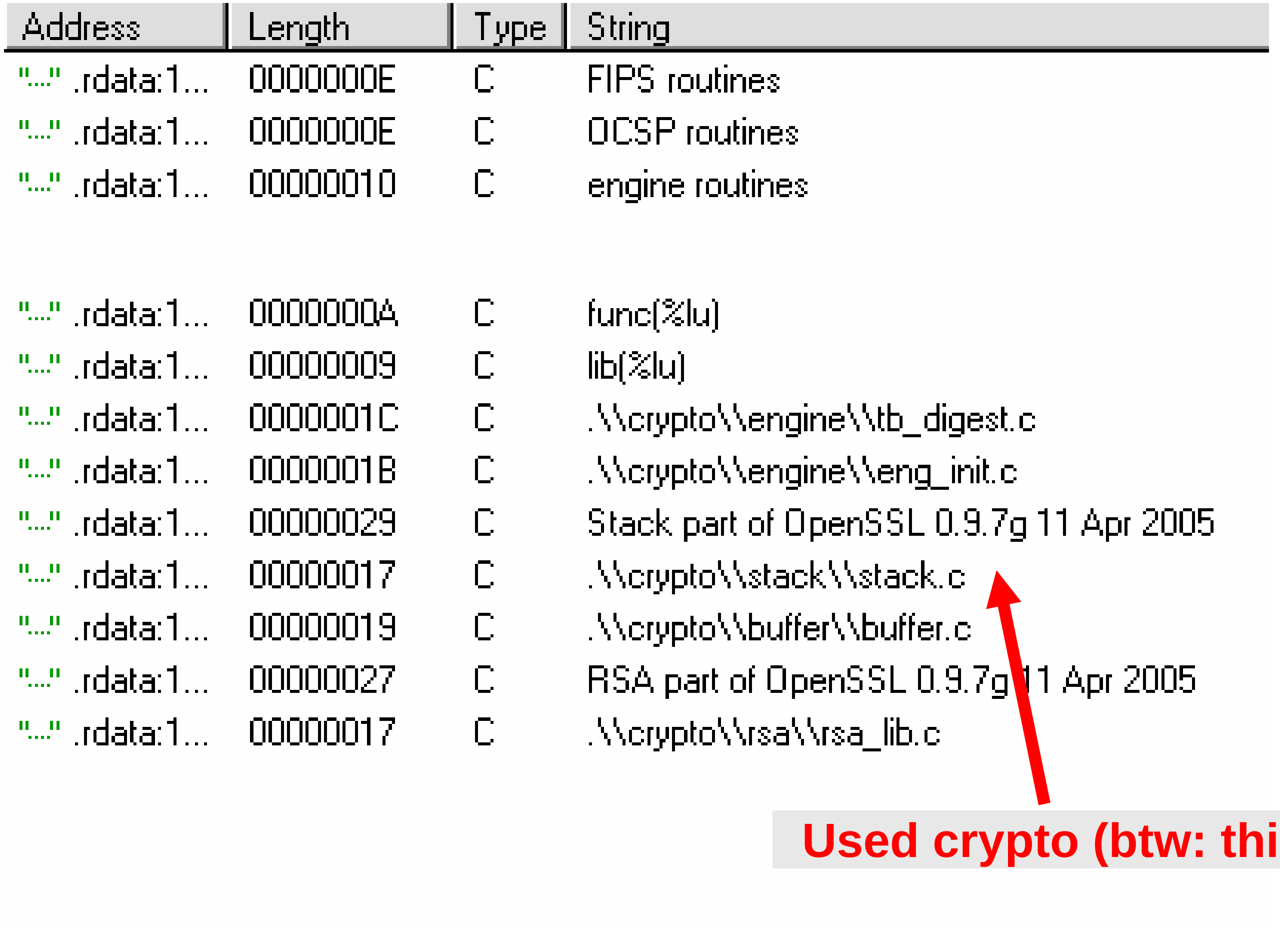
File: "C:\Dokumente und Einstellungen\droecher\Eigene Dateien\01 ernw\50 Research\10 NAC\ P: 22 D: 22 M: 0



```
C:\Daten\MyExploits\NAC\Reversing and Docs\nac-captures\healthy-1\2

80 12 00 18       0A FD 95 DA EF DE 12 77 80 03   ¬....ý"ÚïÞ.w¬.
00 0C EF DE 12 77 1E 26 BD 57 3D 6E 14 B1 80      ..ïÞ.w.&½W=n.±¬
      01 00 04 F6 D3 27 A3                        ....öÓ'£
```

```
C:\Daten\MyExploits\NAC\Reversing and Docs\nac-captures\healthy-1\3

      00 13 00 19 0A FD 95 DB F6 D3 27 A3 80 03   .....ý"ÛöÓ'£¬.
00 0C EF DE 12 77 1E 26 BD 57 3D 6E 14 B1 80 02   ..ïÞ.w.&½W=n.±¬.
00 05 01 01 00 05 01                              .......
```

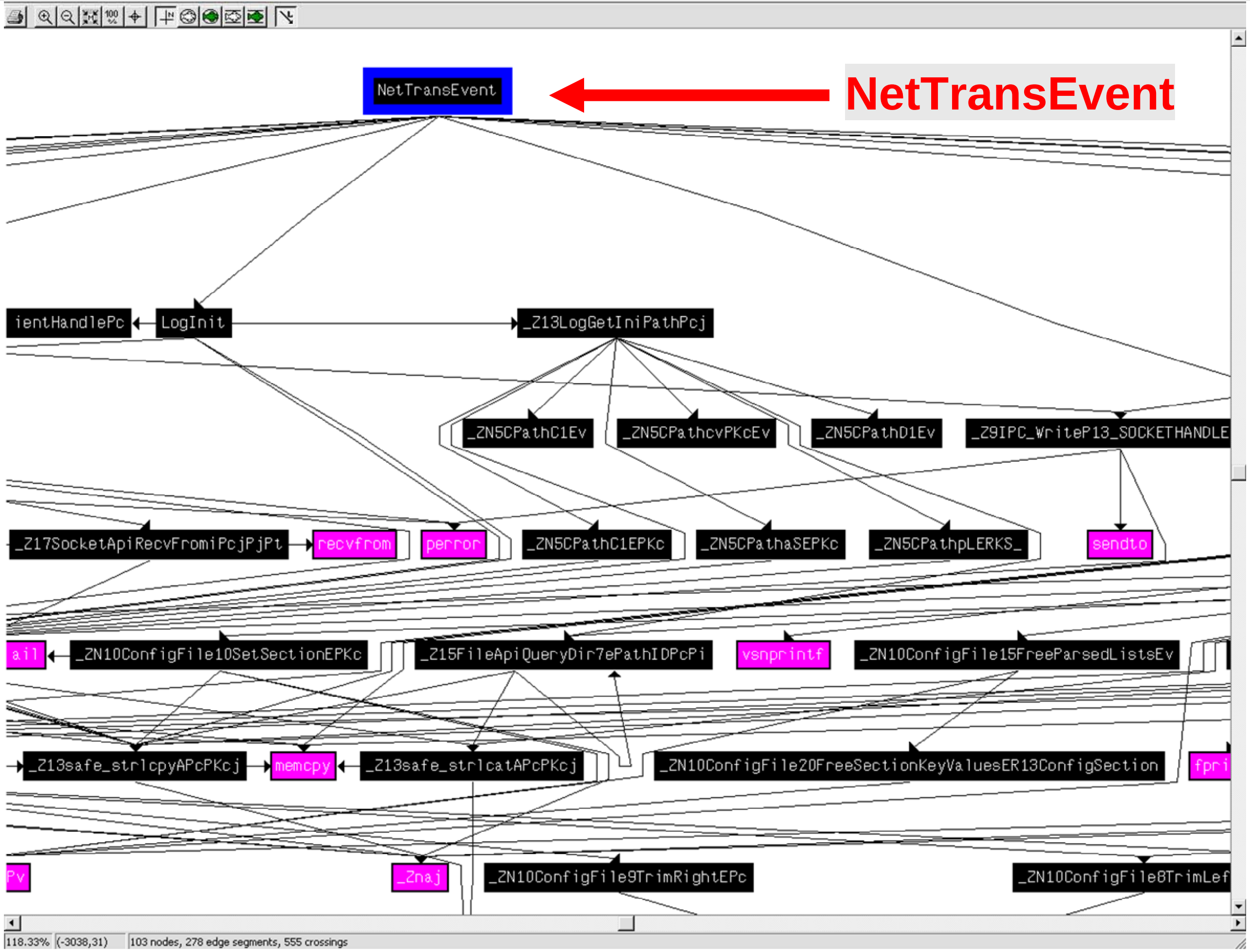
RE of the CTA – 1: Used Crypto



Address	Length	Type	String
"..." .rdata:1...	0000000E	C	FIPS routines
"..." .rdata:1...	0000000E	C	OCSP routines
"..." .rdata:1...	00000010	C	engine routines
"..." .rdata:1...	0000000A	C	func(%lu)
"..." .rdata:1...	00000009	C	lib(%lu)
"..." .rdata:1...	0000001C	C	.\\crypto\\engine\\tb_digest.c
"..." .rdata:1...	0000001B	C	.\\crypto\\engine\\eng_init.c
"..." .rdata:1...	00000029	C	Stack part of OpenSSL 0.9.7g 11 Apr 2005
"..." .rdata:1...	00000017	C	.\\crypto\\stack\\stack.c
"..." .rdata:1...	00000019	C	.\\crypto\\buffer\\buffer.c
"..." .rdata:1...	00000027	C	RSA part of OpenSSL 0.9.7g 11 Apr 2005
"..." .rdata:1...	00000017	C	.\\crypto\\rsa\\rsa_lib.c

Used crypto (btw: this version is vulnerable)

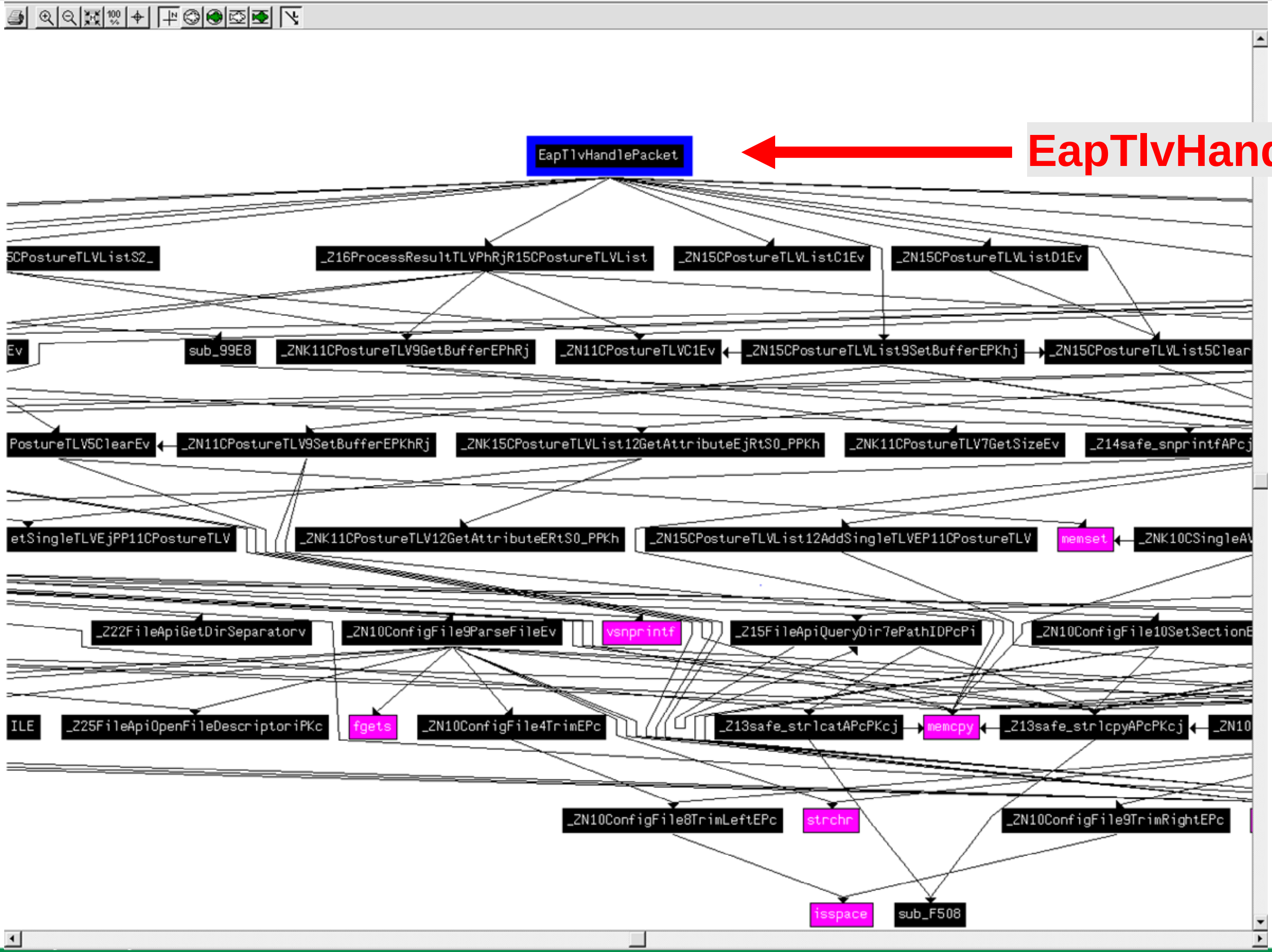
RE of CTA – 1: Core Function



118.33% (-3038,31) 103 nodes, 278 edge segments, 555 crossings

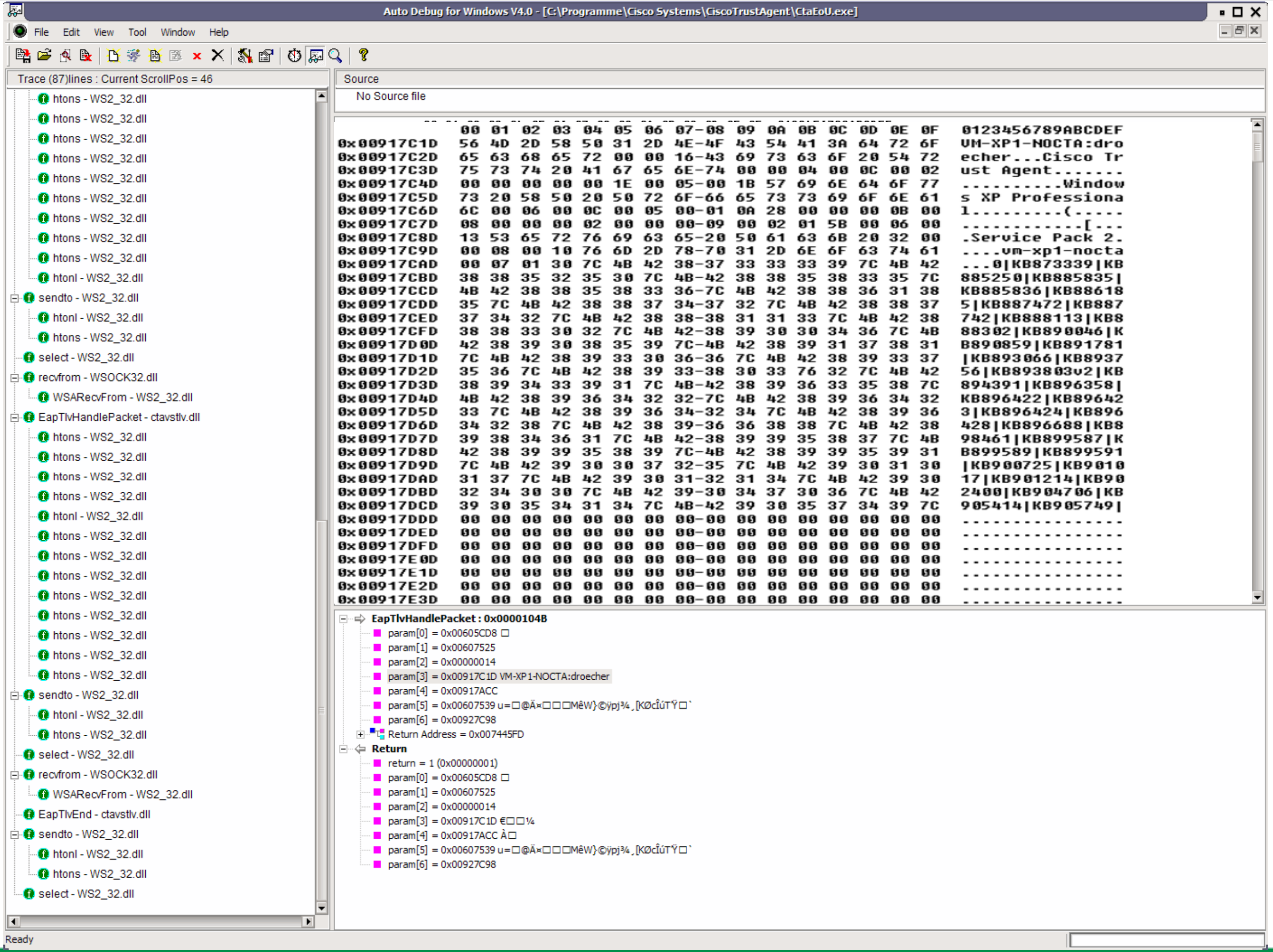
NetTransEvent

RE of CTA – 2: Core Function


EapTlvHandlePacket
5CPostureTLVListS2_
_Z16ProcessResultTLVPhRjR15CPostureTLVList
_ZN15CPostureTLVListC1Ev
_ZN15CPostureTLVListD1Ev
Ev
sub_99E8
_ZNK11CPostureTLV9GetBufferEPhRj
_ZN11CPostureTLVC1Ev
_ZN15CPostureTLVList9SetBufferEPKhj
_ZN15CPostureTLVList5Clear
PostureTLV5ClearEv
_ZN11CPostureTLV9SetBufferEPKhRj
_ZNK15CPostureTLVList12GetAttributeEjRtS0_PPKh
_ZNK11CPostureTLV7GetSizeEv
_Z14safe_snprintfAPcj
etSingleTLVEjPP11CPostureTLV
_ZNK11CPostureTLV12GetAttributeERtS0_PPKh
_ZN15CPostureTLVList12AddSingleTLVEP11CPostureTLV
memset
_ZNK10CSingleA\
_Z22FileApiGetDirSeparatorv
_ZN10ConfigFile9ParseFileEv
vsnprintf
_Z15FileApiQueryDir7ePathIDPcPi
_ZN10ConfigFile10SetSectionE
ILE
_Z25FileApiOpenFileDescriptoriPKc
fgets
_ZN10ConfigFile4TrimEPc
_Z13safe_strlcatAPcPKcj
memcpy
_Z13safe_strlcpyAPcPKcj
_ZN10
_ZN10ConfigFile8TrimLeftEPc
strchr
_ZN10ConfigFile9TrimRightEPc
isspace
sub_F508

EapTlvHandlePacket

Function Hooking into EapTlvHandlePacket



Auto Debug for Windows V4.0 - [C:\Programme\Cisco Systems\CiscoTrustAgent\CtaEoU.exe]

Trace (87)lines : Current ScrollPos = 46

```
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htonl - WS2_32.dll
⊟ sendto - WS2_32.dll
    htonl - WS2_32.dll
    htons - WS2_32.dll
  select - WS2_32.dll
⊟ recvfrom - WSOCK32.dll
    WSARecvFrom - WS2_32.dll
⊟ EapTlvHandlePacket - ctavstlv.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htonl - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
    htons - WS2_32.dll
⊟ sendto - WS2_32.dll
    htonl - WS2_32.dll
    htons - WS2_32.dll
  select - WS2_32.dll
⊟ recvfrom - WSOCK32.dll
    WSARecvFrom - WS2_32.dll
  EapTlvEnd - ctavstlv.dll
⊟ sendto - WS2_32.dll
    htonl - WS2_32.dll
    htons - WS2_32.dll
  select - WS2_32.dll
```

Source

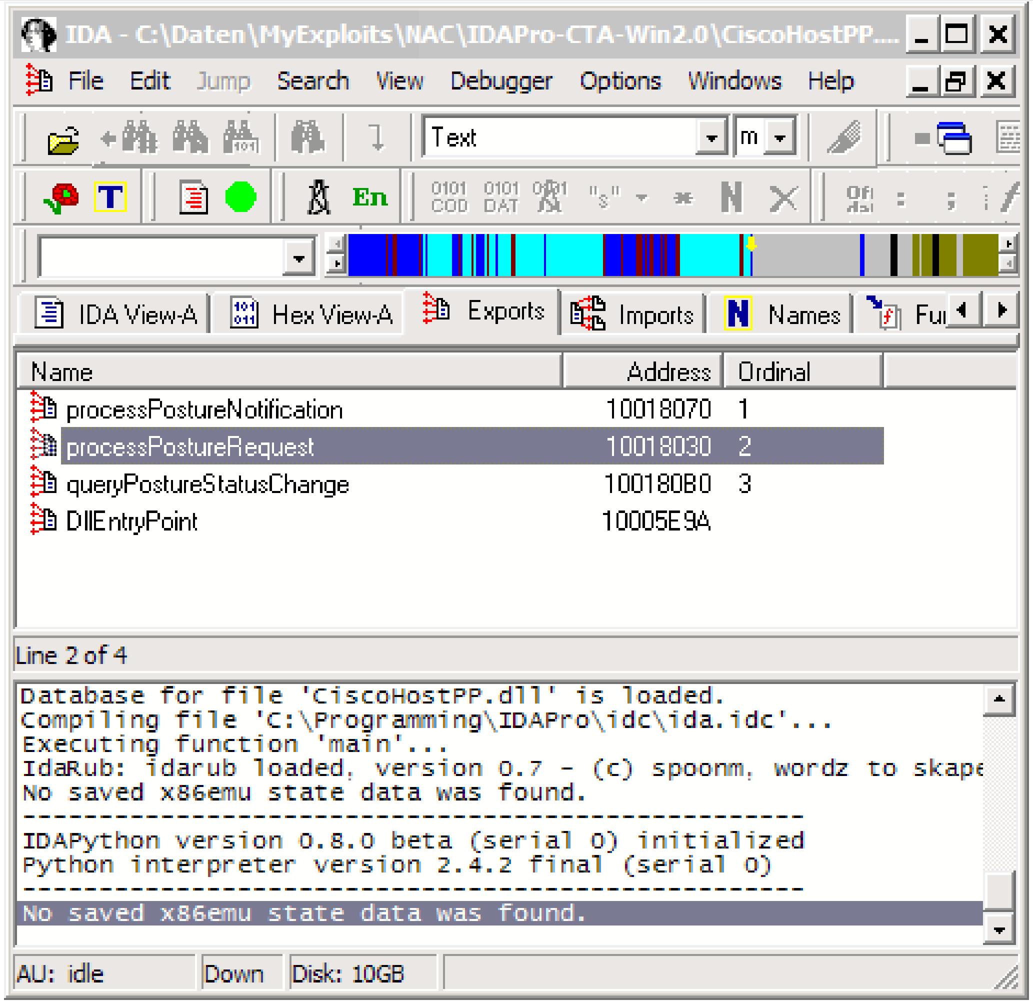
No Source file

```
             00 01 02 03 04 05 06 07-08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
0x00917C1D   56 4D 2D 58 50 31 2D 4E-4F 43 54 41 3A 64 72 6F   VM-XP1-NOCTA:dro
0x00917C2D   65 63 68 65 72 00 00 16-43 69 73 63 6F 20 54 72   echer...Cisco Tr
0x00917C3D   75 73 74 20 41 67 65 6E-74 00 00 04 00 0C 00 02   ust Agent.......
0x00917C4D   00 00 00 00 00 1E 00 05-00 1B 57 69 6E 64 6F 77   ..........Window
0x00917C5D   73 20 58 50 20 50 72 6F-66 65 73 73 69 6F 6E 61   s XP Professiona
0x00917C6D   6C 00 06 00 0C 00 05 00-01 0A 28 00 00 00 0B 00   l.........(.....
0x00917C7D   08 00 00 00 02 00 00 00-09 00 02 01 5B 00 06 00   ............[...
0x00917C8D   13 53 65 72 76 69 63 65-20 50 61 63 6B 20 32 00   .Service Pack 2.
0x00917C9D   00 08 00 10 76 6D 2D 78-70 31 2D 6E 6F 63 74 61   ....vm-xp1-nocta
0x00917CAD   00 07 01 30 7C 4B 42 38-37 33 33 33 39 7C 4B 42   ...0|KB873339|KB
0x00917CBD   38 38 35 32 35 30 7C 4B-42 38 38 35 38 33 35 7C   885250|KB885835|
0x00917CCD   4B 42 38 38 35 38 33 36-7C 4B 42 38 38 36 31 38   KB885836|KB88618
0x00917CDD   35 7C 4B 42 38 38 37 34-37 32 7C 4B 42 38 38 37   5|KB887472|KB887
0x00917CED   37 34 32 7C 4B 42 38 38-38 31 31 33 7C 4B 42 38   742|KB888113|KB8
0x00917CFD   38 38 33 30 32 7C 4B 42-38 39 30 30 34 36 7C 4B   88302|KB890046|K
0x00917D0D   42 38 39 30 38 35 39 7C-4B 42 38 39 31 37 38 31   B890859|KB891781
0x00917D1D   7C 4B 42 38 39 33 30 36-36 7C 4B 42 38 39 33 37   |KB893066|KB8937
0x00917D2D   35 36 7C 4B 42 38 39 33-38 30 33 76 32 7C 4B 42   56|KB893803v2|KB
0x00917D3D   38 39 34 33 39 31 7C 4B-42 38 39 36 33 35 38 7C   894391|KB896358|
0x00917D4D   4B 42 38 39 36 34 32 32-7C 4B 42 38 39 36 34 32   KB896422|KB89642
0x00917D5D   33 7C 4B 42 38 39 36 34-32 34 7C 4B 42 38 39 36   3|KB896424|KB896
0x00917D6D   34 32 38 7C 4B 42 38 39-36 36 38 38 7C 4B 42 38   428|KB896688|KB8
0x00917D7D   39 38 34 36 31 7C 4B 42-38 39 39 35 38 37 7C 4B   98461|KB899587|K
0x00917D8D   42 38 39 39 35 38 39 7C-4B 42 38 39 39 35 39 31   B899589|KB899591
0x00917D9D   7C 4B 42 39 30 30 37 32-35 7C 4B 42 39 30 31 30   |KB900725|KB9010
0x00917DAD   31 37 7C 4B 42 39 30 31-32 31 34 7C 4B 42 39 30   17|KB901214|KB90
0x00917DBD   32 34 30 30 7C 4B 42 39-30 34 37 30 36 7C 4B 42   2400|KB904706|KB
0x00917DCD   39 30 35 34 31 34 7C 4B-42 39 30 35 37 34 39 7C   905414|KB905749|
0x00917DDD   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917DED   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917DFD   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917E0D   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917E1D   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917E2D   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x00917E3D   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

```
⊟ EapTlvHandlePacket : 0x0000104B
    param[0] = 0x00605CD8 □
    param[1] = 0x00607525
    param[2] = 0x00000014
    param[3] = 0x00917C1D VM-XP1-NOCTA:droecher
    param[4] = 0x00917ACC
    param[5] = 0x00607539 u=□@Ä¤□□□MêW}©ÿpj¾¸[KØcÎúTŸ□`
    param[6] = 0x00927C98
  ⊞ Return Address = 0x007445FD
⊟ Return
    return = 1 (0x00000001)
    param[0] = 0x00605CD8 □
    param[1] = 0x00607525
    param[2] = 0x00000014
    param[3] = 0x00917C1D €□□¼
    param[4] = 0x00917ACC À□
    param[5] = 0x00607539 u=□@Ä¤□□□MêW}©ÿpj¾¸[KØcÎúTŸ□`
    param[6] = 0x00927C98
```

Ready

RE of Plug-In 1: Exported Functions


IDA - C:\Daten\MyExploits\NAC\IDAPro-CTA-Win2.0\CiscoHostPP....
File Edit Jump Search View Debugger Options Windows Help
Text
IDA View-A
Hex View-A
Exports
Imports
Names
Fu

Name	Address	Ordinal
processPostureNotification	10018070	1
processPostureRequest	10018030	2
queryPostureStatusChange	100180B0	3
DllEntryPoint	10005E9A	

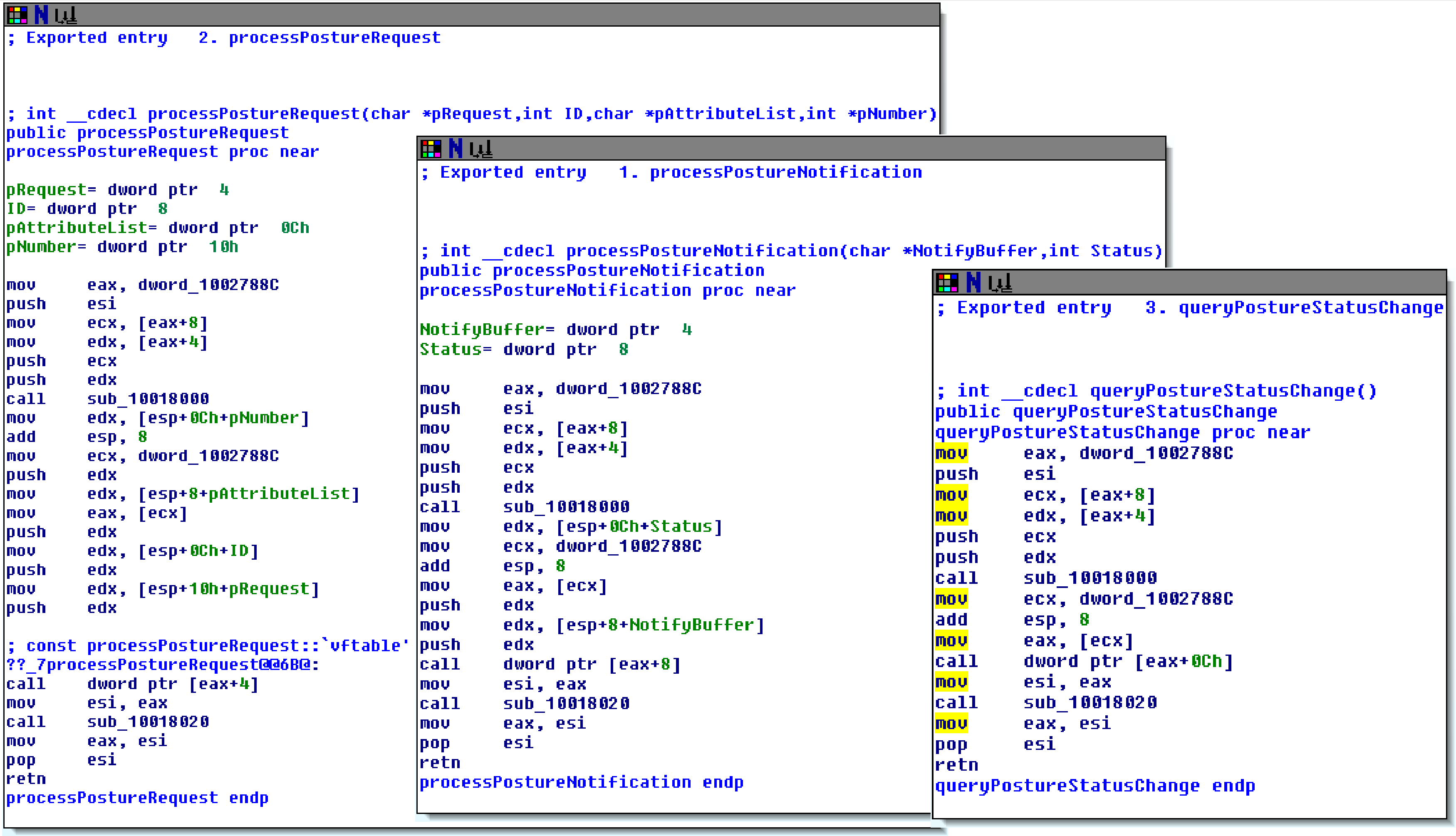
Line 2 of 4

```
Database for file 'CiscoHostPP.dll' is loaded.
Compiling file 'C:\Programming\IDAPro\idc\ida.idc'...
Executing function 'main'...
IdaRub: idarub loaded, version 0.7 - (c) spoonm, wordz to skape
No saved x86emu state data was found.
---------------------------------------------------------
IDAPython version 0.8.0 beta (serial 0) initialized
Python interpreter version 2.4.2 final (serial 0)
---------------------------------------------------------
No saved x86emu state data was found.
```

AU: idle | Down | Disk: 10GB

RE of Plug-In 2: Exported Functions

ERNW
Wir leben IT-Security.

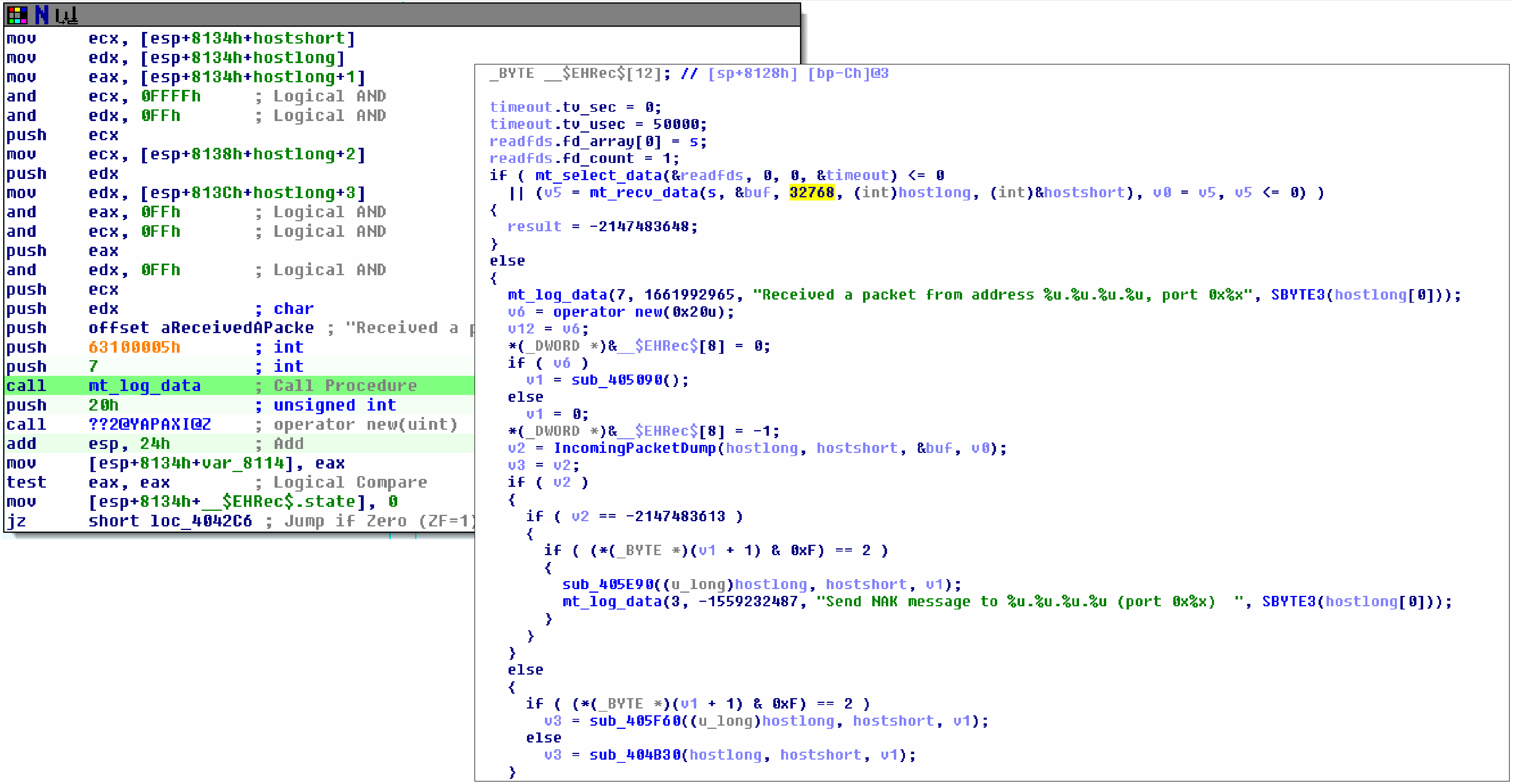


```
; Exported entry   2. processPostureRequest



; int __cdecl processPostureRequest(char *pRequest,int ID,char *pAttributeList,int *pNumber)
public processPostureRequest
processPostureRequest proc near

pRequest= dword ptr  4
ID= dword ptr  8
pAttributeList= dword ptr  0Ch
pNumber= dword ptr  10h

mov     eax, dword_1002788C
push    esi
mov     ecx, [eax+8]
mov     edx, [eax+4]
push    ecx
push    edx
call    sub_10018000
mov     edx, [esp+0Ch+pNumber]
add     esp, 8
mov     ecx, dword_1002788C
push    edx
mov     edx, [esp+8+pAttributeList]
mov     eax, [ecx]
push    edx
mov     edx, [esp+0Ch+ID]
push    edx
mov     edx, [esp+10h+pRequest]
push    edx

; const processPostureRequest::`vftable'
??_7processPostureRequest@@6B@:
call    dword ptr [eax+4]
mov     esi, eax
call    sub_10018020
mov     eax, esi
pop     esi
retn
processPostureRequest endp
```

```
; Exported entry   1. processPostureNotification



; int __cdecl processPostureNotification(char *NotifyBuffer,int Status)
public processPostureNotification
processPostureNotification proc near

NotifyBuffer= dword ptr  4
Status= dword ptr  8

mov     eax, dword_1002788C
push    esi
mov     ecx, [eax+8]
mov     edx, [eax+4]
push    ecx
push    edx
call    sub_10018000
mov     edx, [esp+0Ch+Status]
mov     ecx, dword_1002788C
add     esp, 8
mov     eax, [ecx]
push    edx
mov     edx, [esp+8+NotifyBuffer]
push    edx
call    dword ptr [eax+8]
mov     esi, eax
call    sub_10018020
mov     eax, esi
pop     esi
retn
processPostureNotification endp
```

```
; Exported entry   3. queryPostureStatusChange



; int __cdecl queryPostureStatusChange()
public queryPostureStatusChange
queryPostureStatusChange proc near
mov     eax, dword_1002788C
push    esi
mov     ecx, [eax+8]
mov     edx, [eax+4]
push    ecx
push    edx
call    sub_10018000
mov     ecx, dword_1002788C
add     esp, 8
mov     eax, [ecx]
call    dword ptr [eax+0Ch]
mov     esi, eax
call    sub_10018020
mov     eax, esi
pop     esi
retn
queryPostureStatusChange endp
```

Hex-Rays Decompiler



```
mov     ecx, [esp+8134h+hostshort]
mov     edx, [esp+8134h+hostlong]
mov     eax, [esp+8134h+hostlong+1]
and     ecx, 0FFFFh      ; Logical AND
and     edx, 0FFh        ; Logical AND
push    ecx
mov     ecx, [esp+8138h+hostlong+2]
push    edx
mov     edx, [esp+813Ch+hostlong+3]
and     eax, 0FFh        ; Logical AND
and     ecx, 0FFh        ; Logical AND
push    eax
and     edx, 0FFh        ; Logical AND
push    ecx
push    edx              ; char
push    offset aReceivedAPacke ; "Received a p
push    63100005h        ; int
push    7                ; int
call    mt_log_data      ; Call Procedure
push    20h              ; unsigned int
call    ??2@YAPAXI@Z     ; operator new(uint)
add     esp, 24h         ; Add
mov     [esp+8134h+var_8114], eax
test    eax, eax         ; Logical Compare
mov     [esp+8134h+__$EHRec$.state], 0
jz      short loc_4042C6 ; Jump if Zero (ZF=1)
```

```
_BYTE __$EHRec$[12]; // [sp+8128h] [bp-Ch]@3

timeout.tv_sec = 0;
timeout.tv_usec = 50000;
readfds.fd_array[0] = s;
readfds.fd_count = 1;
if ( mt_select_data(&readfds, 0, 0, &timeout) <= 0
  || (v5 = mt_recv_data(s, &buf, 32768, (int)hostlong, (int)&hostshort), v0 = v5, v5 <= 0) )
{
  result = -2147483648;
}
else
{
  mt_log_data(7, 1661992965, "Received a packet from address %u.%u.%u.%u, port 0x%x", SBYTE3(hostlong[0]));
  v6 = operator new(0x20u);
  v12 = v6;
  *(_DWORD *)&__$EHRec$[8] = 0;
  if ( v6 )
    v1 = sub_405090();
  else
    v1 = 0;
  *(_DWORD *)&__$EHRec$[8] = -1;
  v2 = IncomingPacketDump(hostlong, hostshort, &buf, v0);
  v3 = v2;
  if ( v2 )
  {
    if ( v2 == -2147483613 )
    {
      if ( (*(_BYTE *)(v1 + 1) & 0xF) == 2 )
      {
        sub_405E90((u_long)hostlong, hostshort, v1);
        mt_log_data(3, -1559232487, "Send NAK message to %u.%u.%u.%u (port 0x%x)  ", SBYTE3(hostlong[0]));
      }
    }
  }
  else
  {
    if ( (*(_BYTE *)(v1 + 1) & 0xF) == 2 )
      v3 = sub_405F60((u_long)hostlong, hostshort, v1);
    else
      v3 = sub_404B30(hostlong, hostshort, v1);
  }
```

Hex-Rays Decompiler

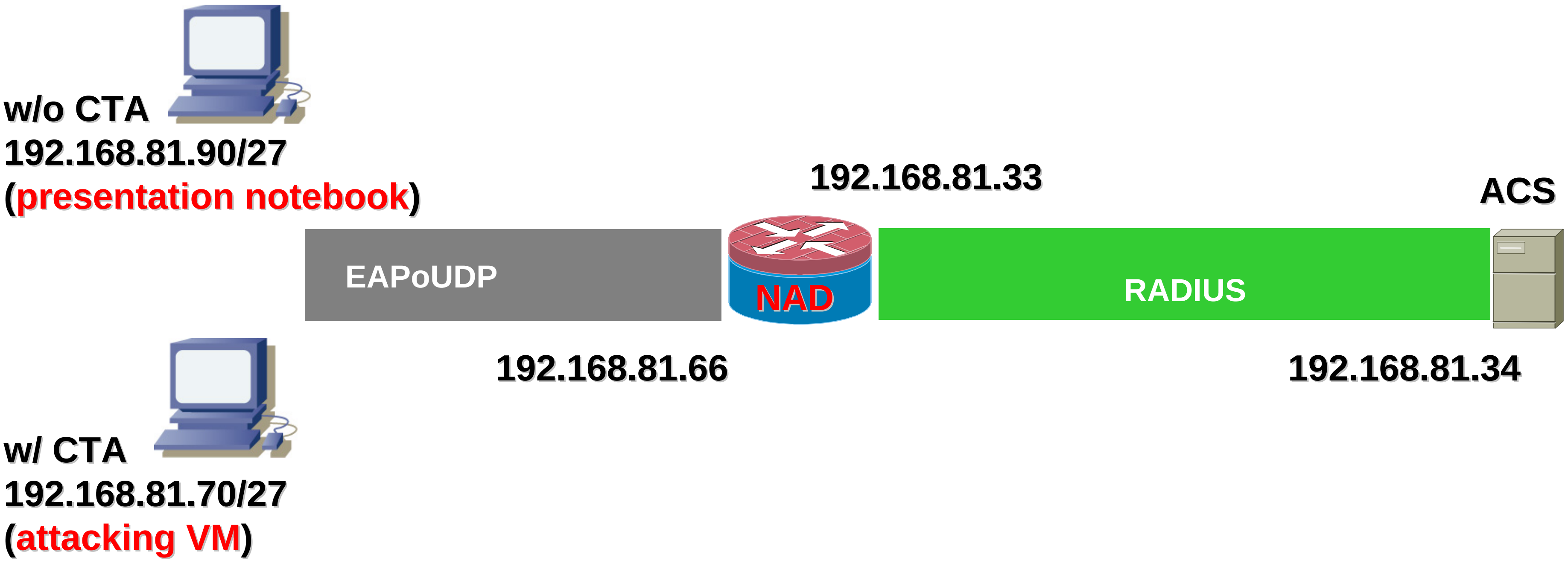
- First Decompiler that produces more than crap
- Build by Ilfak Guilfanov (think IDAPro ☺)
- Actually in Beta State (but already impressing)
- Will be released as commercial Addon for IDA
- Planned: API to support Decompiler Plugins like Vulnerability Analyzer and others
- Planned: Type and Function Prototype Recovery
- Planned: Assembler Knowledge not needed anymore
- Further Information at www.hexblog.com
- Thanks to Ilfak for the Beta Version ☺

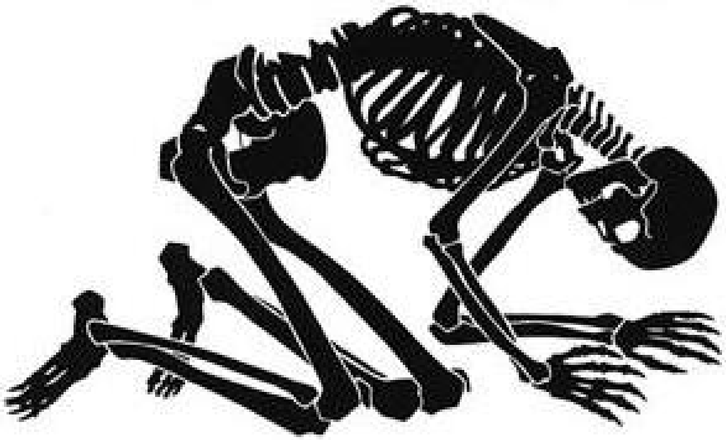
Quick Summary…

- **A lot of stuff learned so far…**
  - What is used
  - How it works
  - How it interoperates
  - Where to start hacking it
- **So now its…**

SHOWTIME

Showtime Setup





ERNW
Wir leben IT-Security.

Thank’s for your patience

Time left for `questions & answers` ?

You can always drop us a note at:
droecher@ernw.de
mthumann@ernw.de