Network Security 19: Practical VPNs

i.g.batten@batten.eu.org

Practical VPNs

- We’ve talked about VPNs in principle
- We’ve talked about IPsec in detail
- What are the real protocols?

Practical VPNs

- SSL VPN
  - OpenVPN
- L2TP + IPsec
- “Cisco style” IPsec VPN

SSL VPN

- Covers several alternatives
  - “Client-less” systems provide access to http resources via an SSL connection between the browser and the VPN server
  - Richer systems use the same connection and a light-weight, often Java, client which does tunnelling

Client Less

- You can argue this is not really a VPN
    - User connects to https://sslvpn.my.com, authenticating once
    - Server presents various corporate resources, all behind sslvpn.my.com (and its certificate)
    - Connection is re-used, and resources are fetched over it, even if they aren't really https-enabled
        - Effectively, the SSL VPN server is a reverse proxy

Client Less Problems

- Causes all corporate resources to appear to come from a single domain as far as browser security is concerned
  - sslvpn.my.com/mail
  - sslvpn.my.com/expenses
- Massive potential for cross-site scripting, fun and games with cookies, etc, if any one of the resources is compromised.

This is fixable

- With care, and a lot of assurance, you could divide corporate resources into sub-domains, so that browser security sees them as distinct
- If you are smart and determined enough to do this, you are smart and determined enough to do something better

SSL Forwarding

- Client downloads a Java app (or similar), which runs in the browser
- Granted elevated privileges in order to listen on 127.0.0.1:port
- Applications talk to 127.0.0.1:port and are tunnelled to server:port.
- Like ssh port forwarding for people who can’t cope with ssh port forwarding

OpenVPN

- Uses authentication mechanisms from SSL, so leverages OpenSSL (used as a library) functionality
- Runs over TCP or UDP so is OK for firewalls, and has proxying extensions
- But fully-featured packet-based VPN which looks like an interface to the computers involved

OpenVPN

- Client requires openvpn client (userspace) and “tun” or “tap” drivers (kernel-side, now shipped with almost every Unix-alike including OSX, Linux and Solaris).
- Server requires suite of daemons, plus similar kernel support.
- Modern versions can do 6-in-6, 6-in-4 and 4-in-6 as well as 4-in-4.
- Implementations available for assorted routers, and commercial virtual appliances.

OpenVPN v SSL VPN

- OpenVPN requires client software in all situations, and installing that software can be an adventure (the iOS stuff is particularly annoying, as it sits outside the OS VPN framework)
- SSL VPN can run clientless, although that has problems
- SSL VPN clients self-install from target address, and in some cases do not need admin password.

OpenVPN Benefit

- OpenVPN can be configured to retain a copy of the **exact** certificate presented by the server
- Means attacker who can forge signed certificates with arbitrary subjects cannot perform MitM
- SSL VPNs usually just use certificates in the normal way, so attacker who can get a certificate in the right name can pose as server

L2TP

- L2TP = Layer 2 Tunnelling Protocol
  - Successor to L2FP (Cisco) and PPTP (Microsoft)
- Sends packets as UDP (so passes through NAT) containing tunnelled data
- Commonly used to tunnel PPP

PPP

- Point to Point protocol
- Derived from HDLC
- Mechanism for sending IP packets down serial lines and other point to point (note: not peer to peer) links
  - Replaced SLIP (Serial Line IP) as the protocol of choice for modems
- Also used for tunnelling elsewhere, cf. PPPoE, PPPoA.

Why PPP?

- PPP has its own authentication mechanism, permitting username/password login as part of setting up a connection
- Also has mechanisms for negotiating MTU, IP Addresses, etc
  - Hence use by ISPs
  - Incoming connections usually handed to a Radius server which authenticates and issues IP numbers

PPP over L2TP

- Raw PPP frames can be encapsulated into L2TP frames
- The arrive at the other end and are removed from the encapsulation
- Overall effect is as though there were a piece of wire between the two points, carrying PPP

Security?

- PPP has no encryption
- L2TP has no encryption (PPTP did, but it was rubbish).
- That looks like a problem, doesn’t it?
- IPsec to the rescue

L2TP/IPsec

- L2TP appears to the network as a flow of UDP packets between the client and the server
- This is ripe for securing with IPsec
- Usual method is using pre-shared keys, but certificates can be used

L2TP/IPsec

- Presented to users as a “group secret” and then their own username and password
  - NB: **this is the only thing protecting all users against a MITM which will yield their personal credentials.**
- Reality (and getting it working on Mikrotik routers!) is rather more complex
- Group Secret is a pre-shared key for IKE (so certificates can be used instead, for the very keen)

L2TP/IPsec

- Client talks IKE to server to establish session keys, secured with group secret common to all users of the VPN server
    - Some VPN client software goes to great lengths to hide this secret from users
- Server and client establish an SPI for L2TP packets
- Client makes PPP connection over L2TP to server, protected by IPsec

L2TP/IPsec

- Group key provides (some) confidence client is talking to the right server, and that the connecting client is not just some random machine on the Internet
- Privacy comes from Diffie-Hellman negotiation of a session key (forward secrecy in event of later compromise of secret)
- PPP login/password proves user is valid and allows per-user profiles (and can use OTP)
- PPP login is protected by IPsec confidentiality

L2TP/IPsec

- Widely available, standard on Windows, Android, OSX and iOS. Components usually present for other systems (although can be complex to set up, as involve merging PPP and IPsec)
- Lots of Appliances available
- Tends to use IKE aggressive mode, hence needs pre-establishment of crypto suite in use (normally 3DES).
  - Can be difficult in heterogeneous environments

“Cisco” IPsec

- Instead of complex stack of PPP over L2TP over IPsec, why not just use IPsec tunnel mode?
- Answer: lack of standardised authentication

IPsec XAuth

- Standard IPsec authentication mechanisms include pre-shared secrets and certificates of various sorts.
- Very much aimed at host-to-host security, rather than user-to-host
  - Doesn’t support any sort of two-factor or challenge/response authentication

IPsec XAuth

- Closest analogue to a password, the pre-shared key, is used directly as a key
  - Needs to be long and random, although you can imagine use of a password derivation function, but also…
  - Both sides needs copies in plaintext, which may not be acceptable
  - No easy way to use to support OTP

IPsec XAuth

- Provides mechanism for a sequence of messages passing arbitrary requests (including nonces) and getting arbitrary responses (including hashes)
- All can then be passed to/from a Radius server
- Annoyingly, weakly standardised and full of Proprietary extensions
    - Supported directly in IKEv2, but transition is slow

“Cisco” IPsec

- Historically “Cisco VPN Client”
- Now “Cisco Anyconnect Secure Mobility Client”
    - Bundled with iOS and OSX, presumably others
    - Cisco logo only third-party branding on iPhone; rumoured to be part of deal over iOS v IOS.
- XAUTH supported by IOS (Cisco operating system) and therefore on various Cisco appliances as well as full-feature routers
- Reverse engineered onto racoon and charon (helped by old RFCs) but only for the enthusiastic
- IKEv2 versions are standardised

VPN Deployment

- VPN tends to work on the assumption that the VPN secures the connection, no-one else needs to worry about it
  - SSL VPNs open up a range of cross-site attacks if resources from different trust domains are aggregated
  - All VPNs provide trusted paths deep into the enterprise, to applications that are not secured
  - Two factor, two factor, two factor

VPN Summary

- SSL VPNs work for their problem space, but get messy and complex for arbitrary applications.
- OpenVPN works, but lack of commercial support on client side an issue
- L2TP/IPsec works well and is supported, but complex stack and difficult to diagnose problems in complex networks
- Cisco stuff easy if your clients are supported, almost impossible if they aren't; requires effectively proprietary extensions to IKEv1.
  - But Cisco support, eg, Solaris on SPARC!