PacketFence, the Open Source NAC: What we've done in the last two years

Salivating on NAC secret sauce

Presentation Plan

- What's Network Access Control (NAC)
- The secret sauce
- The Open Source differentiator
- The good and the bad of 2 years as lead developer
- The Future of PacketFence (aka World Domination Roadmap)
- Community bonding!

Who I am

Olivier Bilodeau

- System architect working at Inverse inc
- PacketFence lead developer since 2009
- Teaching InfoSec to undergraduate in Montreal
- ...

  new father, Open Source nuts, enjoying CTFs a lot, android developer, brewing beer

Social stuff

- twitter: **@packetfence** / identi.ca: **@plaxx**
- delicious: **plaxxx** / linkedin: **olivier.bilodeau**

What's Network Access Control (NAC)

NAC elevator pitch

NAC: Network Access (or Admission) Control

- Authentication

  Map usernames to IP addresses (or MAC addresses)

- Admission

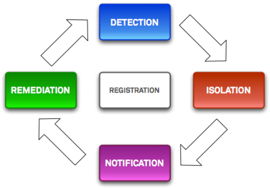
  Allow, partially allow or deny users or devices

- Control

  Watch for unauthorized stuff

  - Including: Outdated AV, patch-level, scanning corporate servers, spreading malware, ...

Know **who** is **using your network** and making sure they **behave**



What NAC has become

- Remediation of users

  Crush helpdesk costs by giving users their own path to fix their problems

- Guest management
- Asset/Inventory management
- Simplified access layer configuration

  Reduce network mgmt costs by centralizing decisions on a srv

The secret sauce

The technology

- Mostly Perl some PHP
- Leveraging open source*
- Designed with high-availability in mind

  active-passive clustering

Key design decisions

- Out of band*
- Edge enforcement*
- No Agent
- Web-based captive portal
- Listen to everything

Out of band

At first, relying on SNMP Traps*

next slide is about that

- LinkUp / LinkDown events
- MAC Nofication events
- Port-Security (SecurityViolation) events

Then RADIUS-based techniques emerged

- Wireless MAC-Authentication*
- Wireless 802.1X*
- followed by Wired MAC-Auth and 802.1X

Edge enforcement: SNMP Traps based

- Events on network hardware generates Traps
- PacketFence reacts on the traps
- Uses SNMP to authorize the MAC / change the VLAN

  or telnet / ssh if the vendor sucks

  port-sec traps have MACs in them so are best otherwise we need to poll

  port-sec fail last-known state

Protocol Reminders

RADIUS

- key-value based protocol for AAA
- "infrastructure" protocol

Protocol Reminders (contd.)

802.1X

- Extensible Authentication Protocol (EAP) over RADIUS
- Actors
  - Supplicant

    client side software integrated in Win, Linux, OSX now
  - Authenticator

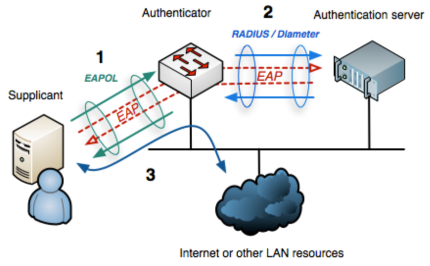
    aka NAS
  - Authentication Server

    NAS is switch / controller, auth srv: FreeRADIUS on PF Server

    explain typical dialog: client speaks to switch/controller with EAPoL (pre-access)

    switch turns around and speak RADIUS with server

    server reacts and send instructions to switch



end-to-end encrypted EAP tunnel is established

several EAP flavors things have mostly settled for PEAP/EAP-MsCHAPv2

switch doesn't have to understand EAP

- Allows to securely share stuff with client (WPA-Enterprise keys)

Protocol Reminders (contd.)

MAC-Authentication

- Simple RADIUS auth with MAC as User-Name
- Concept similar to 802.1X

  infra talks with srv, srv sends instructions
- No strong authentication

  trust based on MAC seen on the wire
- No end-to-end with client

  client doesn't need to "support it"

  not sure what came up first but it feels like a backport of 802.1X

RADIUS CoA (RFC3576)

- Server-initiated

Edge enforcement: RADIUS based

- Access-Accept most request
- Return proper VLAN attribute based on client status
- FreeRADIUS does RADIUS / 802.1X pieces

  full auth incl. NTLM (AD) trough samba
- FreeRADIUS perl extension calls a PacketFence Web Server

  Decision and VLAN returned at this point

  H-A is critical as RADIUS is now a SPOF

The Captive Portal

It provides

- Various authentication mechanism (LDAP, AD, RADIUS, Kerberos, Guests, ...)
- Redirection to the Internet after authentication
- Remediation information to users on isolated devices

The Captive Portal (contd.)

In order to reach the captive portal

- Provide DHCP

  IP to MAC (but we do arp also)
- DNS Blackhole

  In Registration / Isolation VLAN we are the DNS Server

  No matter the request, we return PacketFence's IP
- SSL Redirection
  - Requested URL is re-written to

    http://www.google.com => *https://pf.initech.com/captive-portal*

  WISPr support

Voice over IP

SNMP-based

- Old way: Rely on CDP / Voice VLAN features

  and allow dynamically learned MAC on Voice VLAN

  That's right! No secret here, that's weak!
- New way: handle them as regular devices

RADIUS-based

- MAC-Auth
  - The switch is more important than your device
- 802.1X
  - Some VSA's to control behavior
  - Very few support 802.1X

  Not widespread

Voice over IP (contd.)

Note to pentesters:

- Most want auto-registration of phones
- Accomplished through:
  - MAC Vendor prefix
  - CDP
  - DHCP fingerprints
  - 802.1X MD5 Auth
- Spoof: allowed on the Voice VLAN

  if not worse

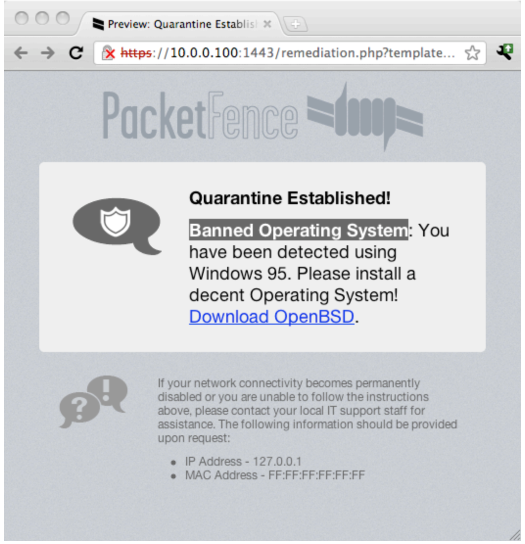
  sometimes Voice VLANs IDs pushed down in DHCP Options!

Quarantine

- On a separate VLAN providing strong isolation
- Triggers:
  - Operating System (based on DHCP fingerprints)

    I talked about those yesterday (FingerBank talk)
  - Browsers (User-Agent string)
  - MAC Vendor
  - Nessus plugin ID (failed scan)*
  - IDS rule triggered*
- Captive portal provides instructions

  Remediation!


Preview: Quarantine Establis
~~https~~://10.0.0.100:1443/remediation.php?template...
PacketFence

Quarantine Established!

Banned Operating System: You have been detected using Windows 95. Please install a decent Operating System! Download OpenBSD.

If your network connectivity becomes permanently disabled or you are unable to follow the instructions above, please contact your local IT support staff for assistance. The following information should be provided upon request:

- IP Address - 127.0.0.1
- MAC Address - FF:FF:FF:FF:FF:FF

Policy checking and Monitoring

Nessus

- Client-side scanning upon authentication
- Somewhat limited

  little use w/o domain credentials (scan open ports?)
- not free

  the more tests the slower

Snort IDS

- Span your traffic to PacketFence server

  available remote also
- Enable some Snort rules
- Devices violating the rules will be isolated

Network Hardware support

- RADIUS-based is easiest
- SNMP is challenging
  - Little standards (nothing regarding port-security)
  - Most implementation differ (even for the same vendor)
  - Nasty bugs*

PacketFence ZEN

ZEN: Zero Effort NAC

- VMware Virtual Appliance
- Pre-installed
- Pre-configured

Open Source FTW!!

The open source advantage

- Vendor independence

  means we support more hardware brands

  and today's networks are heterogeneous. Also no vendor locking
- Proprietary pricing questionable

  (per IP, per concurrent connections, per AP/Switch...)
- We stay focused and build on top of
  - Usual daemons: Apache, Bind, dhcpd
  - Network services: Net-SNMP, FreeRADIUS
  - Security: snort, iptables
  - 70+ Perl CPAN modules
  - Linux!
- familiar stack

The technology is exposed: users know more and there's less reliance on vendors or contractors

- Security is necessarily not solely based on obscurity

  Defeated proprietary NAC by hardcoding sniffed IP/gateway or pinning ARP

2 years as lead developer

The learning, the bad and the good.

Learned: Most NACs are easy to bypass

To achieve user friendliness or network administrator friendliness one often drops security

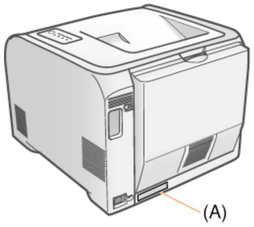
- Per port exceptions (printers, voip, uplinks, etc.): Find them, leverage them
- CDP enabled: Fake being an IP Phone or an infrastructure device
- Real DNS exposed: DNS-tunnel out

Because there is no authentication built-in L2 / L3

- IP Address spoofing
- MAC Address spoofing
- DHCP client spoofing

  Use dhclient with a config file. Spoof VoIP, infrastructure devices to gain access. Could work w/ PacketFence based on config. Well hidden secret though!
- User-Agent spoofing

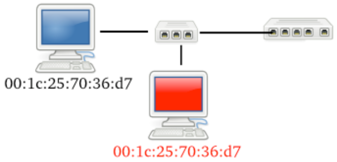
  Spoof a mobile browser, bypass requirement for client Agent. That's how some of the big boys do it..



Learned: Wired 802.1X bypass

802.1X == Port-Based Network **Access** Control

1. Put a hub between victim and switch (prevent port from going down)
2. Wait for victim to successfully authenticate
3. Spoof your MAC with victim's MAC
4. Plug into the hub



Learned: Wired 802.1X bypass

Attack scenarios

1. We keep legitimate client connected
   - Bad: Duplicated MACs on the same segment
   - Good: Original client could re-authenticate if switch asks
2. Replace legitimate client
   - Bad: We won't pass a re-authentication request
   - Good: No network problems (no duplicated MAC on the segment)

Try it out. It works!

Learned: getting into 802.1X is tricky business

- Supplicant support
  - Win: Need to launch a service
  - OS EAP support varies
  - Proprietary supplicant quality / features varies
  - Some hardware begins to impement it
  - Forget about most of them

    Too many things does IP: UPS, slingbox, barcode scanner
- Outside the spec
  - Should a supplicant do DHCP REQUEST or DISCOVER after a new authentication?
  - How should a switch handle multiple supplicant per port?

    Important for VoIP, Virtualization, etc.
  - Unified MAC-Auth + 802.1X configuration tricky

    Timing issues on reboot (dot1x times out, MAC-Auth kicks in)

Learned: Wired 802.1X on Mac OSX is buggy

After 802.1X re-authentication and a VLAN change (through RADIUS VLAN Attributes)

- OSX does unicast DHCP REQUEST to its previous DHCP Server (instead of DISCOVER)
- Does 3 attempts with 1 minute delays between them
- Then resort to a broadcasted DHCP DISCOVER

A "correct" implementation does

- 3 unicast DHCP REQUESTS in a row
- Waits 2-3 seconds for replies
- Then resort to a broadcasted DHCP DISCOVER

Noteworthy

- They had the same issue on wireless but they fixed it in 10.6.5
- We filed a bug report, provided requested information and haven't heard back since

Learned: Network vendor fragmentation

- VLAN assignment through SNMP
- Port-Security
  - Named differently
  - Implemented differently (per VLAN, per port, per port-VLAN)
  - SNMP access inconsistent
- RADIUS-based enforcement
  - Wired MAC-Authentication has many many names

    MAC-Auth Bypass aka MAB (Cisco)

    MAC-based authentication (HP)

    NEAP (Nortel)

    Netlogin (Extreme Networks)

    MAC RADIUS (Juniper))
  - 802.1X's grey areas are all implemented differently
  - RADIUS Change of Authorization (RFC3576) not so supported...

- Newer stacks favor Web Services and only provide read-only SNMP

Fortunately the situation on the wireless side is better

Learned: Network vendors firmwares quality

- Regressions...
- Weird coincidence? Same bugs implemented by different vendors
- PacketFence: I think there's a bug here. Vendor: oh, right! it doesn't work using CLI but it does work with the Web GUI
- Scale issues

  some implement the security table in MAC table. makes everything slower on large L2 VLANs

Learned/rant: Network vendor closeness

I know some people aren't going to agree with this but...

- All vendors hold tight on their issue trackers
- Most vendors hold tight on their firmware
- Some vendors hold tight on their documentation

Learned: Almost nobody does infrastructure authentication

- Asking a user to install/select a CA to authenticate the infrastructure is too much
- Asking the admins to push a GPO with the proper configuration is too much
- Isn't WPA2/Enterprise enough they say?

All the infrastructure to teach the user how to configure themselves can be sent over an open SSID in HTTPS but even then they just don't care! They want youtube, now!

The bad

- First installation step: Disable SELinux
- Too short release cycles for a 'core infrastructure' piece of software
- No nmap integration :(
- External code contributors are scarce
- Pretty much CentOS/RHEL only

The good: Development Process / Infrastructure

- Fully automated smoke tests
- Automated nightly packages promoted to the website (snapshots)
- Stable branches (2.2, trunk) vs feature branches
- All the work is directly public. No internal magic or big code dumps.

The good: Usability++

- Re-organized and simplified documentation
- Simplified installation
- Simplified upgrades
- Default VLAN management technique covers a lot of use cases

The good: Enterprise++

- Web Administration users rights
- Out of the box support for routed environments
- 64 bit support
- Fancy guest workflow support
  - Email activation
  - Hotel-style Access codes
  - Remote pre-registration
  - Approval by a sponsor
  - SMS authentication
  - ...

The good: Performance++

- 1.8.5: ~10x MAC-Auth / 802.1X performance gain
- 1.9.0: Avoiding object re-creation and spawning shell commands (impact not measured)
- 1.9.1: 23x faster captive portal
- 2.2.0: Automatic Apache number of child tweaking based on system memory
- 2.2.1: Reduced by 550% RADIUS round-trip time on environment with lots of network devices

The good: Technology++

- Web Services support for network hardware management
- New architecture for RADIUS-based access using Web Services
  - Strongly decouples RADIUS from PacketFence infra
  - Allows tiered deployment: many local "dumb" FreeRADIUS boxes with a central PacketFence server
  - Multi-site local RADIUS with caching in case of WAN failure
- Demoed a PacketFence in the cloud on Amazon EC2 (Remote RADIUS, local OpenVPN)
- Making in-line and out-of-band work at the same time on the same server

Cool hacks: Proxy Bypass

Bypassing client-side proxy settings

- The problem
  - Browser tries to reach the proxy
  - Proxy doesn't exist in registration / isolation VLANs
  - We rely on the browser to present information to the user
  - We rely on user IP to identify him
  - Worse, SSL through a proxy is done with a CONNECT end-to-end tunnel
- The solution
  - A Squid proxy
  - Squid's URL Redirectors makes sure that all hits are redirected to the captive portal
  - Squid's SSL Bump will terminate CONNECT requests
  - No SSL errors since we bump using the real captive portal cert
  - and everything is still encrypted up to the PacketFence server

Cool hacks: Javascript network access detection

- The problem:
  - Enabling network access delay is unpredictable (OS, switch, browser, ...)
  - Avoid a fixed value otherwise everyone waits for slower
  - Browsers don't like changing IPs / DNS and still run javascript code
- The solution:
  - Turn off DNS prefetching (with HTTP Header)
  - Hidden <img> tag with an onload callback
  - Periodically inject a src that points to an image hosted on a 'registered' VLAN
  - Once the image successfully load, the callback is called and we redirect the user to its original destination

Our World Domination Roadmap

Short-term

- In-line mode to ease support of legacy network hardware (now in beta!)
- reduced complexity
- RADIUS Accounting / Bandwidth monitoring
- NAP / Statement of Health client-side checks
- RADIUS CoA (RFC3576)
- ACL / QoS assignment with RADIUS
- VPN support
- Debian / Ubuntu support

Long-term

- Active-Active clustering support
- nmap / OpenVAS integration
- Making this stuff "Click Next-Next-Next" easy to install
- Rewrite the Web Administration interface

  would get rid of the php

Research topics

- IF-MAP support
- Open source multi-platform client-side agent
- Trusted Computing Group's Trusted Network Connect (TNC)

Community bonding!

This is where we beg for help..

- Network hardware vendors
  - Contact us we want to support your hardware!
- Security software vendors
  - We want to integrate with your IDS, Netflow analyzer, IPS, Web filter, etc. but we need licenses...
- Developers
  - Low barrier to entry: It's all Perl!
- Audit our [web] code. We know there are issues. Help us find and fix them!
- Become users!
- We would **love** to see more businesses/consultants deploying PacketFence for their customers on their own!

That's it

I hope you enjoyed! See you in the debriefing room.

twitter: **@packetfence** / identi.ca: **@plaxx**

delicious: **plaxxx** / linkedin: **olivier.bilodeau**

References

- PacketFence
  - Project Website, http://www.packetfence.org
  - Source Code Repository, http://mtn.inverse.ca
  - Issue tracker, http://www.packetfence.org/bugs
- 802.1X
  - Wikipedia: 802.1X, http://en.wikipedia.org/wiki/IEEE_802.1X
  - An Initial Security Analysis of the IEEE 802.1X Standard, http://www.cs.umd.edu/~waa/1x.pdf
  - Mitigating the Threats of Rogue Machines — 802.1X or IPsec?, http://technet.microsoft.com/en-ca/library/cc512611.aspx
- Research
  - Cisco NAC: No Agent for iOS, http://www.cisco.com/en/US/docs/security/nac/appliance/support_guide/agntsprt.html#wp125743
- Proxy Bypass
  - Feature ticket, http://www.packetfence.org/bugs/view.php?id=1035
  - Squid's SSL Bump, http://www.squid-cache.org/Doc/config/ssl_bump/
  - Squid's Redirectors, http://wiki.squid-cache.org/Features/Redirectors
- Important projects
  - FreeRADIUS, http://freeradius.org/
  - Net-SNMP, http://www.net-snmp.org/
  - The others you already know about
- Tools
  - yersinia: Comprehensive LAN attack tool, http://www.yersinia.net/
  - iodine: IP over DNS tunneling, http://code.kryo.se/iodine/