PacketFence

By Russ McRee – ISSA member, Puget Sound (Seattle), USA Chapter

Prerequisites

VMWare or CentOS Linux platform – installation methodology described is specific to ZEN VMWare appliance

Supported switch to support VLAN isolation and a dot1q trunk

An old boss of mine always found a way to blame the vast majority of security-related problems on "the fuzzy neural network behind the keyboard." Yep, users; what would our lives be without them? There are a plethora of ways, methods, and manners with which to protect your critical assets and networks from said users, amongst them Network Access Control (NAC). A variety of commercial NAC solutions are offered; you may also have heard or read discussions regarding the nuance between Cisco's Network Admission Control and Microsoft's Network Access Protection. As such solutions are proprietary and have costs associated with them, we'll steer clear of any debate and discuss an outstanding free and open source (FOSS) solution, as is the *toolsmith* norm.

PacketFence[1] is a fully supported (tiered support, bronze to platinum, is available, as well as consultation hours or full deployment services)) NAC system that is in use in financial, education, engineering, and manufacturing sectors, to mention a few. It's used right here in my own backyard at Seattle Pacific University and is considered a trusted and indispensable resource.[2] PacketFence sports a robust feature set including a captive-portal for registration and remediation, centralized wired and wireless management, 802.1x support, Layer-2 isolation of problematic devices, as well as integration with both Snort IDS and Nessus.

PacketFence is supported and maintained by Inverse, a Montreal-based firm. PacketFence development is helmed by Olivier Bilodeau, Inverse's System Architect. Olivier provided me with an update on PacketFence news and developments as I prepared for this article.

As his is an open source company generating its revenue via services, the PacketFence roadmap is strongly influenced by customer demand and as such sometimes moves in different direction than the public roadmap.

Expect the following in the next major release, 3.0 (likely mid-August):

- Completely re-designed Captive Portal
- Guest API that, with little effort, allows a lot of different guest management workflows (email confirmation, self-registration, pre-registration, SMS confirmation, hotel-style code generation, etc.)

According to Olivier, the Guest API has been in the making for more than a year in a separate branch and now they think it's ready to be merged in for the pending 3.0 release. Inverse has also been working on other features that may or may not make it to 3.0 including:

- Ability to run PacketFence in out-of-band and inline mode at the same time.[3]
- Pushing ACLs/Roles per device/user on the edge. This would allow for more granular control over who has access to what, without VLAN management overhead, and enforced at the edge instead of at the firewall. They are also experimenting with applying QoS the same way.
- Integration with RADIUS Accounting to track bandwidth consumption per user and potentially enforce bandwidth usage restrictions.

Olivier wants to reinforce that all the development is conducted in the open, released under the GPL, and committed directly to public repositories,[4] so all the features mentioned above are available (although at varying degrees of completion). Nightly snapshots are built directly from the trunk branch and several tests are run on it to make sure it meets a certain quality. This makes it very easy to preview upcoming PacketFence features in a staging environment or a VM. Inverse maintains PacketFence installations where there are more than a thousand switches and even more access points, including several customers who crossed the "25,000 devices handled by PacketFence"line in the last two years or so. As such, Olivier strongly affirms that PacketFence competes with big brand commercial offerings both in cost, features, and scalability.

1 http://www.packetfence.org/home.html.

2 http://www.packetfence.org/tour/testimonials.html.

3 http://www.mail-archive.com/packetfence-devel@lists.sourceforge.net/msg00248.html.

4 http://mtn.inverse.ca/.

FIGURE 1 – PacketFence Nodes

In addition to keeping an eye on PacketFence.org, you're encouraged to subscribe to the PacketFence Twitter feed to stay abreast updates on what they're are working on: @packetfence.

Finally, if you're going to be in Las Vegas for Defcon 19, be sure to check out Olivier's presentation, *PacketFence, The Open Source NAC: What We've Done in the Last Two Years.*[5]

Installation/configuration

First, PacketFence is supported by rich documentation; I'll only cover some details that were directly applicable to my experience setting it up in the *toolsmith* lab.

Core to a sound PacketFence installation is a supported switch. Refer to *PacketFence Version 2.2.1 Network Devices Configuration Guide*[6] for device specifics. I used a Cisco Catalyst 3548 XL for this effort, but said switch is rather dated. The Catalyst 3548 XL does not support 802.1X, the preferred port-based Network Access Control method, so I was limited to MAC detection/isolation and was not able to push PacketFence nearly to the extent I would have liked to.

Inverse includes a VMWare appliance, aptly named ZEN (Zero Effort NAC),[7] perfect for folks wishing to assess a fully installed and preconfigured version of PacketFence 2.2.1 when on a tight time line. Again, *PacketFence ZEN version 2.2.1 Installation Guide*[8] is there to support your efforts in full.

Note: the ZEN VMWare appliance version of PacketFence requires a 64-bit capable system.

For those planning dedicated installations, the recommended distributions are RHEL 5 or CentOS 5 for which Inverse offers a yum repository.[9]

If you intend to investigate PacketFence via the ZEN appliance, you'll need to configure your supported switch as follows:

```
VLAN 1 - management VLAN
VLAN 2 - registration VLAN (unregistered
devices will be put in this VLAN)
VLAN 3 - isolation VLAN (isolated devices will
be put in this VLAN)
VLAN 4 - MAC detection VLAN (empty VLAN: no
DHCP, no routing, no nothing)
VLAN 5 - guest VLAN
VLAN 10 - "regular" VLAN
```

Refer to the IP and subnet table on page 7 of the *ZEN Installation Guide* for network configurations per VLAN; DHCP and DNS services are provided by PacketFence ZEN.

5 https://www.defcon.org/html/defcon-19/dc-19-speakers.html#Bilodeau2.

6 http://www.packetfence.org/downloads/PacketFence/doc/PacketFence_Network_Devices_Configuration_Guide-2.2.1.pdf.

7 http://www.packetfence.org/download/vmware_appliance_zen.html

8 http://sourceforge.net/projects/packetfence/files/PacketFence%20ZEN/2.2.1/PacketFenceZEN_Installation_Guide-2.2.1.pdf/download.

9 http://www.packetfence.org/download/releases.html.

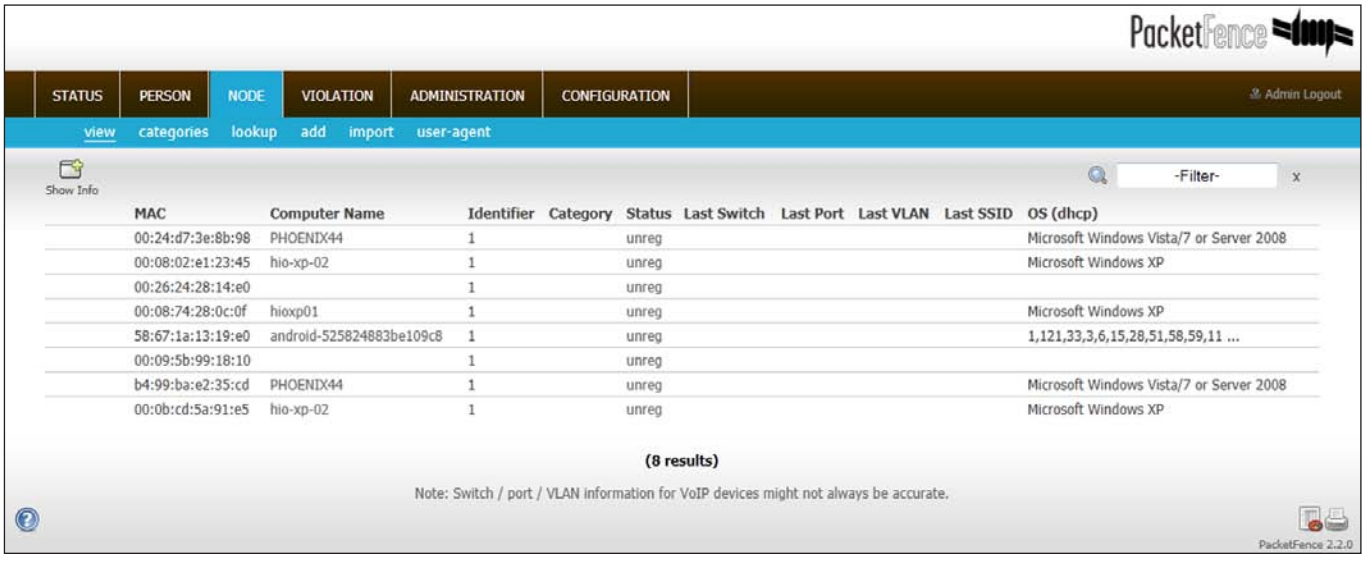
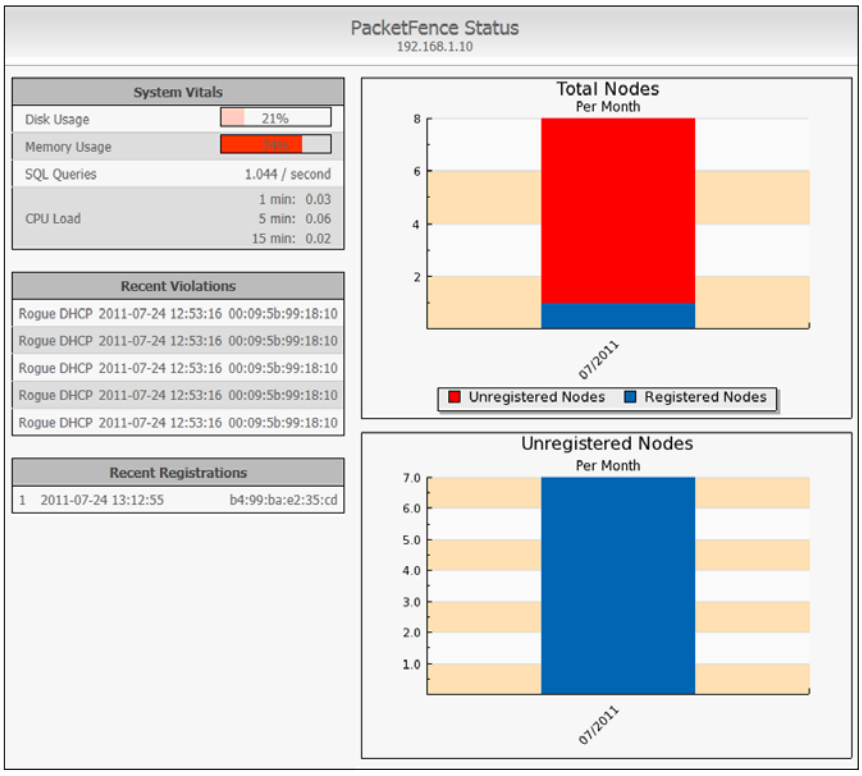


FIGURE 2 – PacketFence flags hacked Nook fingerprint as unknown

FIGURE 3 – PacketFence Status



I set the switch up with an IP address of 10.0.10.2, and on interface f0/48 I defined the port as a `dot1q trunk` (several VLANs on the port) with VLAN 1 as the native (untagged) VLAN as required for the PacketFence (PacketFence) ZEN host.

This is easily done on a Cisco switch as follows:

```
enable
conf term
int fa0/48
switchport trunk encapsulation dot1q
switchport mode trunk
end (Cntrl Z)
```

The ZEN VM appliance is preconfigured to match interfaces to VLANs; just be sure they're all set to bridged under `VM => Settings => Network Adapter`.

If all is configured properly, and your Zen VM appliance is connected to the `dolq trunk` interface, you should be able to browse to https://192.168.1.10:1443 and make use of the PacketFence UI.

Note: You'll discover a slight mismatch between the PacketFence ZEN guide and the network configuration guide where the network configuration guide describes the PacketFence host IP as 192.168.1.5. If you're going with the ZEN installation guidance and using the ZEN VM, the PacketFence VM appliance IP is 192.168.1.10.

The PacketFence work flow exemplifies a network "perfect world" in my opinion. A host that joins the network does so in a limited capacity (no Internet or access) via MAC detection (VLAN4) and is then shunted to registration (VLAN2). Upon successful registration a host continues either as a guest (VLAN5) or an approved system for normal access (VLAN10).

Figure 1 offers a node view including all the unregistered hosts identified by my test instance of PacketFence.

PacketFence includes a database of known fingerprints and user-agents for ready system identification, and will flag unknowns as seen in `Status => Reports`. PacketFence ironically flagged my Barnes & Noble Nook as seen in Figure 2, most likely because I blew the proprietary B & N OS away and loaded it with CyanogenMod 7.1.0 RC1 (Android 2.3.4), which rocks, by the way.

Out of the gate, PacketFence also detected my normal DHCP service and flagged mine as rogue, tagging it as in violation as seen right in the PacketFence Status view depicted in Figure 3.

Legitimate DHCP servers can be added to the pf.conf file preventing alarms for these servers. DHCP is also run in Registration and Isolation VLANs but it's recommended that users to manage their own DHCP servers for the Normal VLANs.

As a security wonk my favorite PacketFence features are, of course, security-related:

1. Detection of abnormal network activity (Snort) where PacketFence defines alerting and suppression coupled with administratively configurable actions.
2. Proactive vulnerability scans (Nessus) conducted during registration, as scheduled, or on an adhoc basis.
3. Isolation for hosts in violation and remediation through a captive portal including logic that distributed the appropriate counsel for violators including:
   - **Banned devices:** "You have been detected using a device that has been explicitly disallowed by your network administrator."
   - **Malware:** "Your system has been found to be infected with malware. Due to the threat this infection poses for other systems on the network, network connectivity has been disabled until corrective action is taken."

Unregistered and/or unmitigated hosts remain in splendid isolation; life is good.

Once a host is registered it can be added to a category; I arbitrarily defined Trusted Hosts via `Nodes => Categories`.

PacketFence offers extensive reporting with output to CSV and the UI, enhanced by extensive filtering. Figure 4 shows registered host with details.

As you build out and experiment, be sure to take a close look at `Configuration` settings. You can define/modify interfac-

FIGURE 4 – Registered Host Report

es, networks, switches, and fine tune the language included in messages distributed to violators trapped in the captive portal.

In conclusion

PacketFence is an outstanding offering and my only regret is not being able to commit more time to testing or pushing its feature set. It left me feeling nostalgic for the days when I was a network systems administrator configuring devices and network security on a regular basis.

Don't limit your thinking specific to PacketFence; while it's a great solution for small/medium business, it really can handle the enterprise as well.

Remember the documentation is extensive. Make use of it to get fully wrapped around ALL of PacketFence's capabilities, then test and deploy.

PacketFence is definitely one of my candidates for "Tool of the Year."

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

Acknowledgements

—Olivier Bilodeau, Inverse System Architect, PacketFence project lead

About the Author

Russ McRee, GCIH, GCFA, GPEN, CISSP, is team leader and senior security analyst for Microsoft's Online Services Security Incident Management team. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ at holisticinfosec dot org.