Multi-platform TNC with Radiator, XSupplicant and libtnc

This white paper discusses the theory and application of Trusted Network Connect endpoint integrity checking in a heterogeneous environment using Radiator, XSupplicant and libtnc.

1.0 Introduction

A perennial problem for network administrators is whether (and how much) to trust devices that connect to their network. Traditionally, access to a wireless network depends only on knowing a valid username and password. And for a wired network, access may only depend on finding an unoccupied network socket.

Further, employees or students might take their laptop home and connect directly to the internet, get infected by a virus or malware, and then bring the infected machine back into the core of their employer's network where the virus has free reign.

Recently a number of efforts have concentrated on this problem of 'endpoint integrity': how to be sure a user's laptop is clean and secure before letting it connect to the corporate network.

One of those efforts is TNC: Trusted Network Connect. TNC is a public specification for a system that checks endpoint integrity during network connection. Endpoints such as user's laptops get checked for valid anti-virus software or other corporate prerequisites before being allowed to connect to the network.

TNC has been developed by Trusted Computing Group (http://www.trustedcomputinggroup.org) a non-profit consortium whose goal is to develop and promote open, vendor-

neutral, industry standard specifications for trusted computing building blocks and software interfaces across multiple platforms.

Products that support TNC are now starting to appear in the market place. However, most TNC enabled products are closed and proprietary, and in all cases only support a limited subset of computing platforms and operating systems. This is a serious problem for administrators who wish to deploy TNC in a heterogeneous (multi-platform and/or multi-Operating System) environment. While most vendors supporting TNC do so on Intel compatible processors with Microsoft Windows, few support it on Linux or Mac OS X, and none on other operating systems or processors in common use. Further, the ability to customize the TNC components is deliberately limited by the proprietary vendors. Source code is never provided. A number of other commentators have remarked on the absence of TNC support on Mac OS X, for example.

The remainder of this white paper discusses an open-source implementation of TNC support for 802.1X clients, in conjunction with a full-source RADIUS server, allowing administrators to roll out and customize TNC support for a much wider range of devices on their network.

2.0 802.1X and XSupplicant

The more limited problem of secure authentication of connections to wired and wireless networks is basically solved now. 802.1X specifies a series of protocols that will securely and reliably carry authentication credentials (such as username and password) from wireless (or wired) computers to, say, a RADIUS server, where the username and password is checked. Using for example EAP-PEAP or EAP-TTLS, users can only connect to the network if their username and password are correct, and dishonest people are unable to crack or sniff the authentication details from the wireless signals. 802.1X can be used to protect both wired and wireless networks, and is now the system of choice for secure authentication of network access.

FIGURE 1. A typical architecture for 802.1X + RADIUS password based access control

802.1X authentication relies on software called the 'supplicant' which runs on the user's computer. The supplicant talks to the wireless access point (or switch in a wired network) and negotiates the new connection, usually sending the username and password over an encrypted tunnel. The wireless access point sends the authentication credentials to a RADIUS server where they are checked and the RADIUS server then tells the access point whether to accept or reject the connection.

There are a number of commercial and free supplicants available for a range of operating systems and processors. Most are proprietary and closed source, such as the one supplied with Windows XP and Vista, or the commercial pay-per-license Odyssey supplicant from Juniper Networks. However, there are no commercial supplicant vendors that support all operating systems and platforms.

There are, however a number of free, open-source supplicants, which can run on a wide range of operating systems and processors. One such supplicant is XSupplicant, developed by the Open1X group. XSupplicant runs on Unix, Linux, Windows, Mac OS X and others, and on a range of processor types, with a range of 802.1X compliant wireless and wired access points. XSupplicant comes with full source under the GPL and the freedom to modify it to suit your needs.

However 802.1X on its own is not sufficient for endpoint integrity checking. For that you need TNC.

3.0 TNC and libtnc

TNC (Trusted Network Connect) is an open specification describing protocols and APIs for enforcing endpoint integrity. Developed by the Trusted Computing Group, it speci-

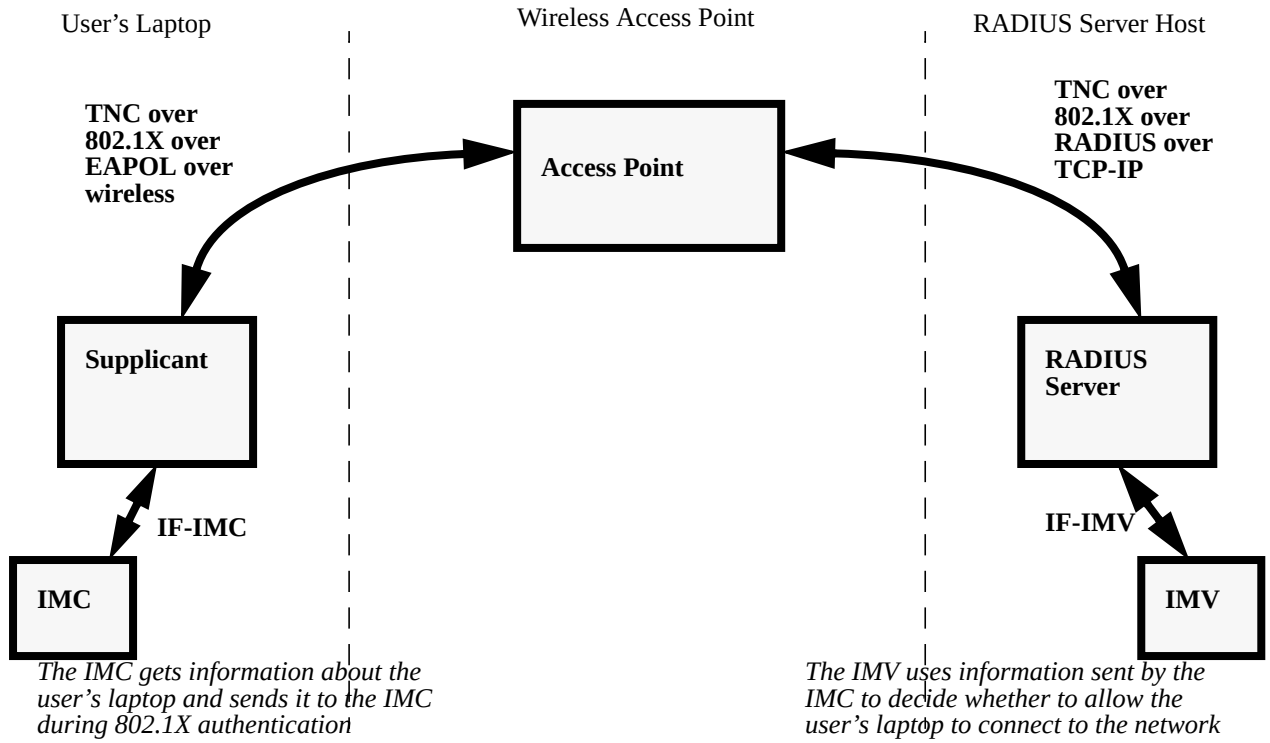
fies how a supplicant, Access Point (or network switch) and a RADIUS server can communicate in order to confirm that the user's laptop is 'clean and secure' by some criteria specified by the network administrator.

TNC is developed by a membership of vendors and independent experts. Open System Consultants' Chief Technologist Mike McCauley is an invited expert on the TNC project group, contributing to the development of the TNC specifications.

In the TNC model, a software module called the IMC (Integrity Measurement Collector) runs on the user's laptop and is accessible by the 802.1x supplicant. On the RADIUS server runs a complementary software module called the IMV (Integrity Measurement Verifier). During network connection, and after the user's password has been checked, the IMV communicates with the IMC on the laptop and checks that the laptop complies with whatever preconditions the IMV requires. Typically the IMV will ask the IMC to confirm that a particular software package (say, an anti-virus package) is valid and up-to-date. Or it might require that a particular operating system version and patch set is installed. Only if the IMC conforms to the requirements will the Access Point be directed to let the user connect to the corporate LAN. Otherwise they might be connected to an isolated 'remediation' LAN where they may be limited to installing software to fix their laptop.

This means that with TNC, only users who have *both* a valid username and password *and* a secure laptop will be allowed to connect to the corporate network.

FIGURE 2. How TNC communicates over 802.1X and RADIUS from laptop to RADIUS server.



libtnc is a open-source library that implements the protocols and APIs specified by TNC. Developed by staff at Open System Consultants and released as free software under the GNU Public License (GPL), libtnc can be used in any open-source application that requires TNC interoperability. libtnc is also available under a commercial license for integrators who do not wish to comply with the GPL.

libtnc includes all the infrastructure required to add TNC support to an application. It includes functions to load IMCs and IMVs dynamically into the application's address space, and to call their TNC specified functions. It also includes skeleton IMC and IMV modules, allowing developers to produce their own custom IMCs and IMVs. libtnc also includes a fully functional, multi-platform, multi-OS IMC that can answer a range of TNC queries about the host it is running on.

Like XSupplicant, libtnc runs on Windows, Linux, Mac OS X and other operating systems and platforms.

One of the open-source projects that uses libtnc is XSupplicant, discussed above. When XSupplicant is built with TNC support and the libtnc library, it can participate in a TNC conversation with a TNC enabled RADIUS server during 802.1X authentication.

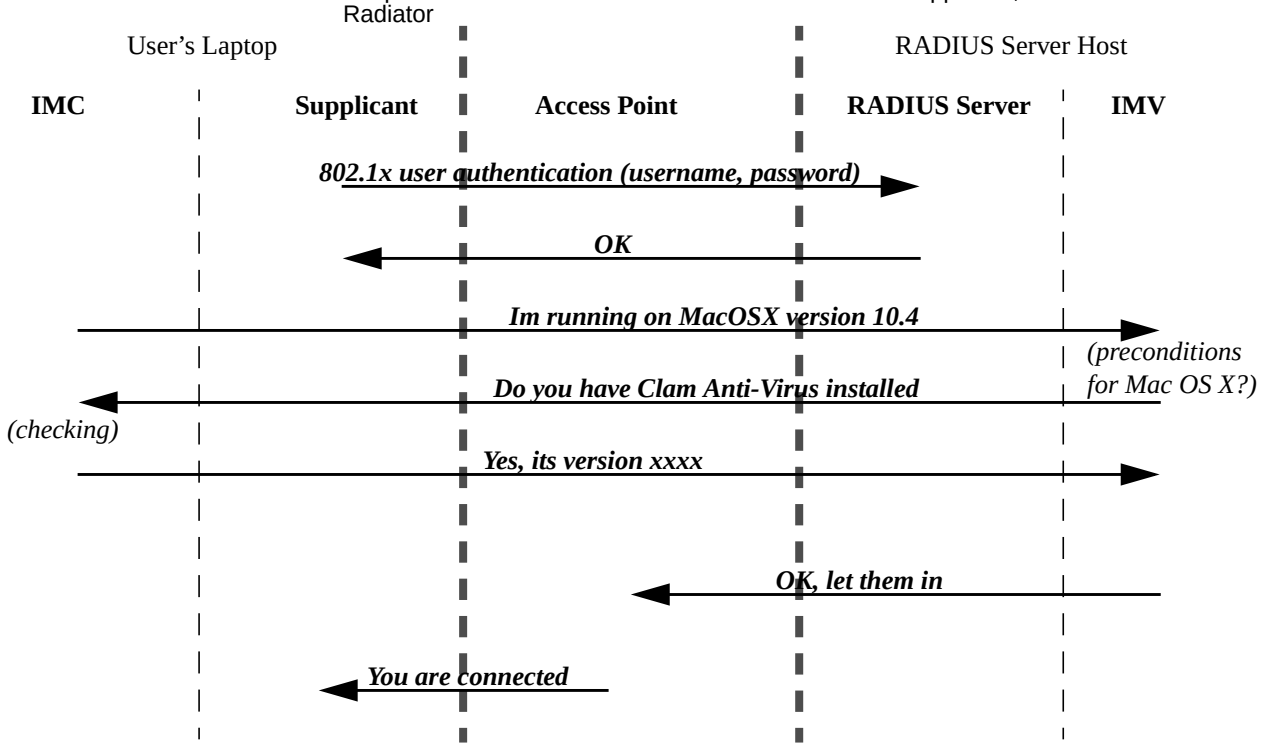
The final link in a TNC deployment is a TNC enabled RADIUS server, such as Radiator.

4.0 Radiator

Radiator is a full-source commercial RADIUS server from Open System Consultants. Open System Consultants has a long history of cooperation with the Open1X group's XSupplicant and other open-source projects, and Radiator interoperates with XSupplicant (with or without TNC support).

Radiator includes support for TNC over 802.1X, and can interoperate with XSupplicant+libtnc running on any platform. Radiator also runs on any platform and can be extensively configured to suit almost any need. Radiator includes a module to interrogate the sample IMC supplied with libtnc, and to accept or reject access based on the answers it gets from the laptop's IMC. Radiator can be configured to specify what conditions a laptop needs to meet before being granted access. These conditions can depend on the laptop operating system type, packages installed, registry values etc.

FIGURE 3. Simplified timeline of a TNC conversation between XSupplicant, Access Point and Radiator



User is now connected to the corporate network and can get on with their work

5.0 Summary

Together XSupplicant+libtnc+Radiator can be used to build a complete TNC enabled access control system in a multi-platform heterogeneous environment, almost out of the box. Furthermore, administrators have at their disposal all the tools they need to build a custom TNC enabled network security system to suit their requirements.

This means that 802.1X+TNC can now be deployed over a wide range of operating systems and platforms, greatly exceeding the potential coverage and customizability of any single TNC vendor.

6.0 References

1. Open1X source code and documentation for XSupplicant
`http://open1x.sourceforge.net/`

2. Trusted Computing Group
   http://www.trustedcomputinggroup.org
3. Trusted Network Connect specifications for TNC etc.
   http://www.trustedcomputinggroup.org/groups/network/
4. Open System Consultants for information about Radiator etc.
   http://www.open.com.au/radiator
5. Joel Snyder discusses TNC and the lack of support on Mac and Linux
   http://www.networkworld.com/reviews/2007/041907-nac-intro.html