GT Apiary: Remote Honeypots
Undergraduate Special Problem, Summer 2010

George Macon
August 4, 2010

a-pi-ar-y n. a place where bees are kept; a collection of beehives.

1 Introduction

Honeypots are an important tool for network security researchers. Traditionally, a network of honeypots, or “honeynet”, would be set up in IP space that was otherwise unused. These honeypots were found by attackers as a result of blind scanning of the IP space (something that will become much more difficult with the introduction of IPv6). While this information is useful, the attacks launched against real companies may be different and more interesting. In order to capture these attacks, honeypots must be set up in corporate public IP space. Since most companies are not engaged in academic research, there is little incentive for them to set up honeypots and analyze the resulting data. The GT Apiary plans to use a virtualized architecture where the companies are given a “turn-key” forwarder that allows academic researchers to set up virtual honeypots in the company’s IP space.

We also analyze a potential problem with this approach: Attackers may discover the true nature of our forwarder and honeypots and stop directing their attacks to them. In order to gain insight into this characterization, we set up a dummy network with one host forwarded over the virtual infrastructure. We expect to see that the latency caused by the encapsulation and decapsulation of the traffic is sufficient to be noticeable from outside the network.

2 Prior Work

The Apiary had some infrastructure as a result of previous work by Master’s student Peter Huang [1]. Huang considers HoneyMole, which is a dedicated project to provide Ethernet over Internet for the purpose of creating remote honeypot forwarders, to be unsupported and buggy. The latest release of HoneyMole occurred in 2008 [2], so this appears to still be the case. His infrastructure used the Layer 3 tunneling capability of OpenVPN and reverse-NAT on the forwarder.
HoneyMole sets up Layer 2 bridges using a custom security layer based on the OpenSSL library. The frames from all foreign networks are trunked onto a single IEEE 802.1q link, and a switch is used to keep the VLANs for each remote network separate. All remote network forwarders talk to the same server.

3 Infrastructure

Both Huang’s approach and HoneyMole’s approach to the infrastructure have merit. For this phase, the good qualities of both approaches were combined. From Huang, we take the concept of using off-the-shelf, well-supported software, and from HoneyMole we take the concept of using a Layer 2 bridge for transparency.

The Apiary uses a virtualized architecture. Our VPN server and corresponding honeypots will exist together on a virtual machine server. For each remote network, we create a new virtual Ethernet to house the honeypots. This provides separation between the various remote networks. The VPN server can act as the honeywall, or something like honeyd can be installed on each honeypot. The exact nature of the honeypots is out of scope for this Special Problem. An architecture diagram is on the next page.

I used the Arch Linux minimal Linux distribution as a base for the bridging. It has a large collection of packages, allowing us to install anything we need, but has a minimal base install that provides high performance and security. In addition to the base packages, the openssh and ntp packages were installed to ease system administration.

For the forwarder and VPN server, Arch’s documentation on setting up OpenVPN Bridging was used. We are using X.509 certificates for authentication. Each remote network has a private instance of the OpenVPN server running on its own port. It also gets a virtual Ethernet interface from the VM server and a Linux Ethernet bridge interface.

On the forwarder and server, iptables is used to “stealth” the box to prevent its appearing to an attacker. For the server, this is straightforward since the bridged interfaces have no IP address configured. This causes the server to appear to the honeypot and the remote network as if it were an Ethernet switch. Since the VPN server may act as the honeywall, iptables can be used to filter the traffic that flows across the bridges. This creates a completely transparent firewall; since it doesn’t have an IP address, it won’t show up on traceroute, but it can still inspect and optionally drop every frame.

On the forwarder, which exists in the IP space of the remote network, stealth is more important. Incoming Ethernet frames that are being bridged pass through the FORWARD chain. Because of this, we apply our filtering on the INPUT chain. With bridging enabled, the host IP address is bound to the bridge interface instead of to the real Ethernet interface. We simply allow packets that are related to established connections and drop all others. We also allow incoming connections to the SSH service for remote management.
4 Honeypot Detection

If attackers can determine that a specific machine is a honeypot, they are likely to stop attacking it. By allowing our honeypots to virtually share a subnet with production machines, we provide a level of defense against this detection, and even if the honeypot is detected, the attackers cannot simply blacklist the entire subnet without also blacklisting “real” targets. Using Layer 2 bridging is another mechanism to prevent detection. Since our concern is attacks coming over the Internet, attackers are unable to see anything below Layer 3.

This leaves a rather large weakness: timing. Since the packets destined for the honeypot must be forwarded over the Internet twice, they will have longer round-trip times that packets to the real hosts on the same subnet. The nmap port-scanning tool was used to collect measurements of a dummy network with one host forwarded through the VPN, but not over the Internet. These timings are shown in Table 1. These values are calculated using the same formula used by TrP for round-trip estimation. The forwarder is 130.207.202.5, and the VPN server is 130.207.202.13. Note that neither of these hosts are listed in the table because they were determined by nmap to be down. The honeypot is 130.207.202.4.

In order to calculate “suspicious” hosts, statistical outliers were calculated as those whose mean or variance were more than the Inter-Quartile Range (1 IQR) away from the median. The median RTT is 3124 µs, and the outliers are 130.207.202.1 and 130.207.202.7. If the threshold is increased to 1.5 IQR, then only 130.207.202.7 is an outlier. This indicates that the VPN, on its own, is not sufficient to cause major timing differences. Since one of the legitimate hosts had a RTT almost twice the median, this seems to suggest that even when forwarding over the Internet, this timing-based characterization will be inconclusive.

5 Future Work

Immediate future work is to further develop the infrastructure with a protected management subnet and a CA to issue certificates for the VPN. This infrastructure may additionally entail the addition of another VPN to allow remote
researchers to use the Apiary. Additionally, forwarders need to be set up in at least one remote location so that data regarding Internet-traversing tunnels may be collected, and to demonstrate isolation of the separate honeypot networks. Further future work is to provide a management interface for the administrators of the host networks and remote maintenance access for the Apiary administrators to maintain the forwarders.

6 Conclusion

A remote honeypot system was configured and seems to be working. Although we were unable to set up the forwarder on a remote network, there doesn’t seem to be a reason to think that the system will not work when set up in that way. We were surprised to see that the VPN did not seriously increase the latency of the system. OpenVPN uses the Blowfish cipher in Cipher Block Chaining mode by default. According to the OpenSSL benchmark, our VPN server is capable of encrypting or decrypting at a rate of 90 MBps to 100 MBps, depending on the block size. This rate will easily saturate a 100 Mbps Ethernet link. The detection will have to come from the additional Internet traversal, a subject for future work.

References


