TORTILLA

TOR… ALL THE THINGS

Jason Geffner
jason@crowdstrike.com
www.crowdstrike.com
CROWDSTRIKE, INC.

Contents

Introduction

In the past, malware research was rather passive. Researchers received malware samples from customers, industry partners, and honeypots, and would analyze the files in a network-isolated environment. These researchers would find indicators of compromise and develop detection signatures for the malware, then move on to the next sample.

Today, malware research often falls under the umbrella of security intelligence research. In addition to analyzing malware samples, researchers now need to actively interact with malicious actors' servers. Whether it's to monitor malicious actors' Command-and-Control servers, download the actors' malware for analysis, or read the actors' blogs in order to harvest actionable intelligence, researchers today are not working in network-isolation.

However, as more and more researchers are now working from home, the notion of exposing one's home IP address to an adversary is rather unsavory, especially when dealing with organized crime syndicates and nation-state adversaries. As such, researchers need a way to anonymously communicate with servers operated by hostile entities.

This whitepaper discusses a new software project named Tortilla, which is designed to allow researchers to easily, safely, and securely use Tor to anonymously communicate over the Internet.

Tor

The Tor Project[1], founded in 2006, oversees development and operation of the Tor software and a global Tor network, which together allow users to anonymously route their traffic through the Internet.

“Tor protects you by bouncing your communications around a distributed network of relays run by volunteers all around the world: it prevents somebody watching your Internet connection from learning what sites you visit, and it prevents the sites you visit from learning your physical location.”[2]

The Tor client works by first obtaining a list of Tor nodes from a Tor directory server. When the client seeks to establish a connection with a destination server, the client picks a random path to the destination server through several Tor nodes and uses onion routing[3] to encrypt the routing information and network data for each hop along the route.

The Tor software is free and open-source, and the Tor network (which is free to use) consists of over 4,000 relay servers[4].

[1] https://www.torproject.org

[2] https://www.torproject.org/docs/faq.html.en#WhatIsTor

[3] http://patentimages.storage.googleapis.com/pdfs/US6266704.pdf

[4] http://torstatus.blutmagie.de/

Tor Browser Bundle

When most people want to use Tor on Windows, they download the Tor Browser Bundle[5] (TBB). The two major components of the TBB are Tor and a modified version of Firefox ESR[6].

Tor runs a SOCKS[7] server, listening locally on TCP port 9050. Tor securely routes incoming SOCKS connections through the anonymizing Tor network. The modified version of Firefox routes all of its DNS and HTTP(S) traffic through this SOCKS server.

Though download and installation of the TBB are relatively easy and indeed sufficient for most basic web surfing needs, the TBB has many shortcomings with regard to overall online anonymity.

Firstly, Firefox is the only browser natively supported by the TBB. This could be an issue for security conscious users, given that Firefox had more than six times the number of vulnerabilities discovered in it over the past two years than did Internet Explorer[8]. Furthermore, plugins (such as Adobe Flash) are not compatible with TBB since TBB cannot redirect plugin-based Internet traffic.

The Tor SOCKS server listening on TCP port 9050 is not restricted solely to Firefox traffic, though. Indeed, any application that is capable of communicating via a SOCKS proxy can be configured to work with Tor, and third-party software such as Privoxy[9] allows HTTP-proxy-aware applications to work with Tor as well. Unfortunately, most software does not support HTTP and/or SOCKS proxying, and even software that does support such proxying doesn't typically support DNS proxying via SOCKS (thus leading to DNS leaks and a loss of confidentiality).

The world needs a more robust solution.

[5] https://www.torproject.org/projects/torbrowser.html.en

[6] http://www.mozilla.org/en-US/firefox/organizations/

[7] http://en.wikipedia.org/wiki/SOCKS

[8] http://secunia.com/?action=fetch&filename=Secunia_Vulnerability_Review_2013.pdf

[9] http://www.privoxy.org

Tortilla Design Goals

Given that the ideal solution will *wrap* Tor around all traffic that can be handled by Tor, this new platform is named Tortilla.

Tortilla consists of five simple design goals:

1. **Transparently route all TCP and DNS traffic through Tor**
   Tor is designed to route IPv4 TCP traffic. It does not handle any other protocols, though it does allow specially formatted DNS lookups to be made via a Tor exit node. Thus, given that Tor can be used for TCP and DNS, Tortilla is designed to transparently route this traffic. In this context, "transparent" routing means that applications need not have native support for Tor or generic SOCKS- or HTTP-proxying. This means that not only can non-Firefox browsers be used (along with plugins), but most any networking applications can be used that utilize TCP and DNS.

2. **Do not allow any network traffic onto the Internet unless it goes through Tor**
   All IPv4 TCP and DNS packets should go through Tor, and all other packets should be dropped. This packet filtering should be at OSI Layer 2, as filtering at Layer 3 or above could allow for some packets to "slip through" to the network card.

3. **Do not require a typical user to install an unfamiliar OS**
   As of July, 2013, Microsoft Windows has a 91.51% market share on desktop computers[10]. While various flavors of Linux are more conducive to intercepting network traffic, Windows is still much more familiar to most desktop users. In order to maximize adoption and minimize the barrier to entry, we do not want to require users to learn a new operating system.

4. **Do not allow malware to circumvent the Tor tunnel to communicate directly with the Internet**
   Ideally, Tortilla's integrity should remain intact even if a user's system is exploited by visiting a malicious website, opening a malicious e-mail attachment, or executing a malicious download. Although standard user mode or kernel mode hooks could be used to programmatically hook and redirect network traffic, malware could disable these hooks and circumvent the Tor-redirection tunnel.

5. **Do not require extra hardware or extra virtual machines (VMs)**
   Extra hardware costs extra money. Extra VMs require extra RAM, which in turn costs extra money. In order to make Tortilla as accessible as possible, it should not have any unnecessary requirements.

[10] http://www.netmarketshare.com/report.aspx?qprid=8&qpcustomd=0

Several solutions already exist that meet some of these design goals, but none meet them all:

Solution	Design Goal					Comments
	1	2	3	4	5	
Hardware-Based Transparent Proxy	✓	✓	✓			Onion Pi[11] and P.O.R.T.A.L.[12] are the most well-known solutions in this category. We assume the user can buy a prebuilt and preconfigured device. Malware is not self-contained because it could connect to a different WiFi network and circumvent the Tor tunnel offered by Onion Pi.
Software-Based Transparent Proxy	✓	✓		?		Tor does not support transparent proxying on Windows since such proxying is implemented via `/dev/pf`. As such, this approach requires a non-Windows gateway (such as Whonix[13]). Furthermore, if the transparent proxy is accessed via a VPN, malware may be able to disable the VPN route to circumvent the Tor tunnel.
Tor Integrated Directly into OS	✓	✓			✓	Tails[14] is Debian-based, not Windows-based.
Winsock Hooks	✓		✓		✓	Winsock hooks (such as the type used by Torcap[15]) can be circumvented by malware.

[11] http://learn.adafruit.com/onion-pi

[12] https://github.com/grugq/portal

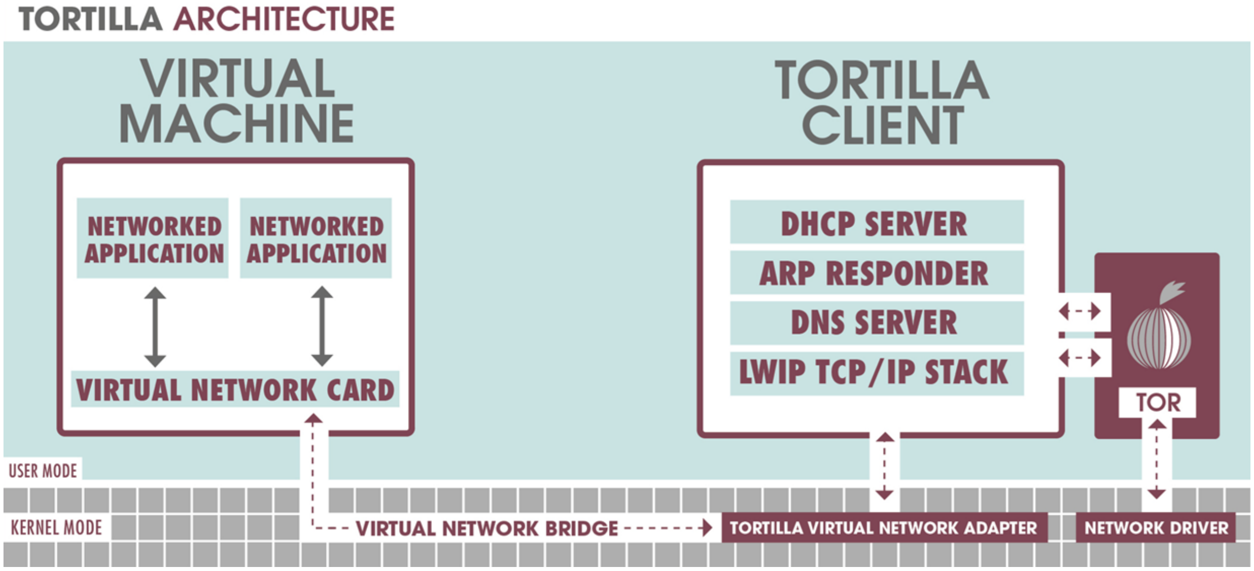
[13] https://whonix.org

[14] https://tails.boum.org/

[15] http://freehaven.net/~aphex/torcap/

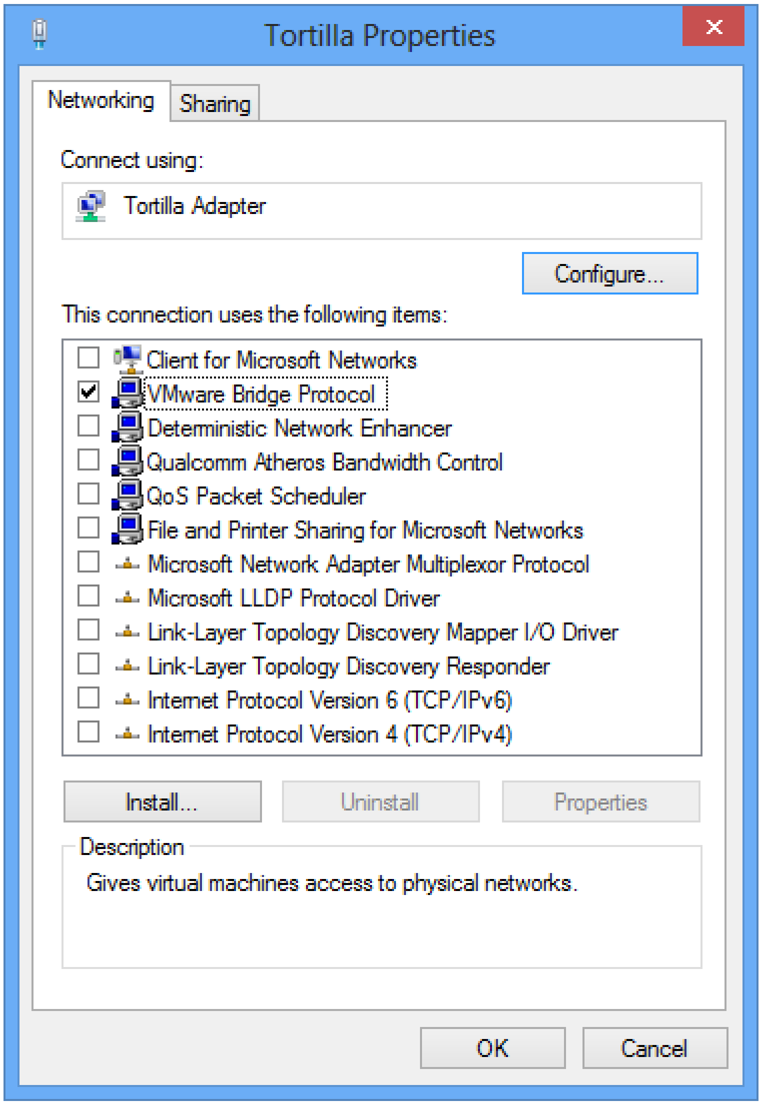
Tortilla Architecture

At a high level, Tortilla's architecture is as follows:



Networked applications are run inside of a VM so that malicious code can't circumvent the Tor tunnel to directly access the Internet and can't discover identifying information on the host system. Since Tortilla's two components—the Tortilla Client and Tortilla Virtual Network Adapter—both run only on the host system, the solution is VM-platform agnostic (VMware, Bochs, Virtual Box, etc.) and guest-OS agnostic.

Tortilla installs a virtual network device named Tortilla Adapter and a corresponding NDIS miniport driver named tortilla.sys. At installation time, Tortilla also disables all network component bindings (clients, services, and protocols) except for that of the Virtual Network Bridge, to ensure that those components on the host OS do not interfere with Tortilla's traffic:



Connect using:

Tortilla Adapter

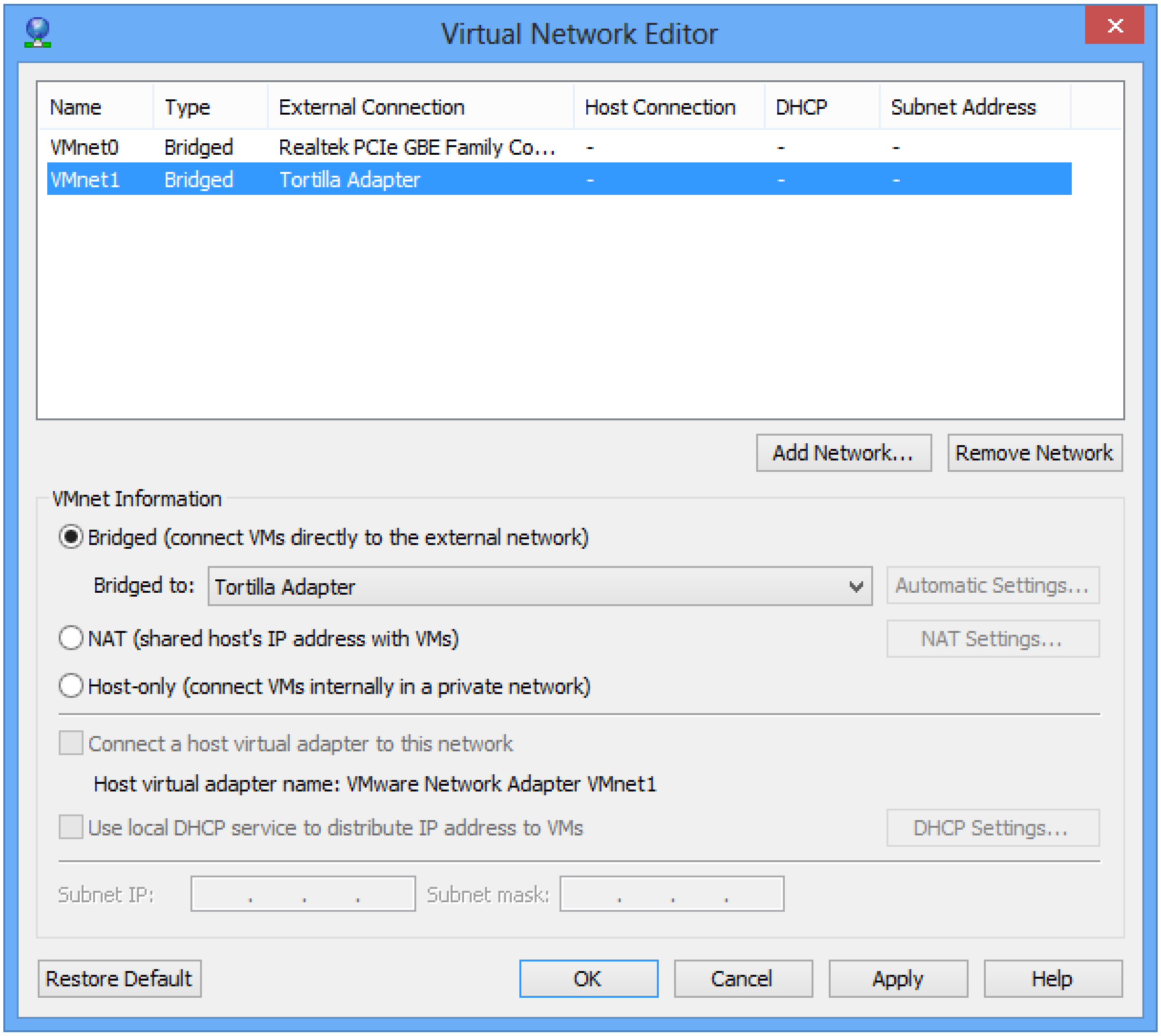
This connection uses the following items:

- □ Client for Microsoft Networks
- ☑ VMware Bridge Protocol
- □ Deterministic Network Enhancer
- □ Qualcomm Atheros Bandwidth Control
- □ QoS Packet Scheduler
- □ File and Printer Sharing for Microsoft Networks
- □ Microsoft Network Adapter Multiplexor Protocol
- □ Microsoft LLDP Protocol Driver
- □ Link-Layer Topology Discovery Mapper I/O Driver
- □ Link-Layer Topology Discovery Responder
- □ Internet Protocol Version 6 (TCP/IPv6)
- □ Internet Protocol Version 4 (TCP/IPv4)

Description

Gives virtual machines access to physical networks.

After the Tortilla Adapter is installed, the user can configure their VM platform to bridge their VM's virtual network card to the Tortilla Adapter:



Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Realtek PCIe GBE Family Co...	-	-	-
VMnet1	Bridged	Tortilla Adapter	-	-	-

VMnet Information

● Bridged (connect VMs directly to the external network)

Bridged to: Tortilla Adapter

○ NAT (shared host's IP address with VMs)

○ Host-only (connect VMs internally in a private network)

□ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet1

□ Use local DHCP service to distribute IP address to VMs

Subnet IP: . . . Subnet mask: . . .

Once bridged, the Tortilla Client receives all OSI Layer 2 network traffic from the VM's virtual network card.

The Tortilla Client handles four different types of packets received from the VM:

- **DHCP Discover/Request Packets**
  The Tortilla Client acts as a simple DHCP server and assigns to the VM client an IP address, IP subnet mask, gateway IP address, and DNS server IP address, all of which are configurable.

- **ARP Request Packets**
  The Tortilla Client responds to all ARP requests (except for requests for the VM client's IP address) with the default Tortilla MAC address `7A:C0:7A:C0:7A:C0` (TACO TACO TACO), which is also configurable.

- **DNS Type A and Type PTR Query Packets**
  The Tortilla Client parses DNS Type A and Type PTR queries, extracts the queried hostname or IP address, connects to the local Tor SOCKS server, and uses the Tor-specific SOCKS command `SOCKS_COMMAND_RESOLVE` or `SOCKS_COMMAND_RESOLVE_PTR`, respectively, to resolve the queried hostname or IP address via Tor. The Tortilla Client then crafts a DNS response packet with the resolved information (or lack thereof) and replies to the VM client.

- **TCP Packets**
  When the Tortilla Client detects a TCP SYN packet from the VM client, it uses the local Tor SOCKS server to try to connect to the destination server via Tor. If the connection is actively refused by the destination server then the Tortilla Client responds to the VM client with a TCP RST packet. If the connection passively fails then the Tortilla Client doesn't respond to the TCP SYN at all, effectively dropping the packet. However, if the Tortilla Client can successfully connect to the destination server via Tor, it hands the connection to the integrated open-source Lightweight TCP/IP[16] (lwIP) stack, which acts to "proxy" TCP sessions between the VM client and the Tor SOCKS server. Tortilla uses a slightly modified version of lwIP which allows it to not only transparently bind to all IP addresses, but to all TCP ports as well.

All other types of packets are ignored and effectively dropped.

[16] http://savannah.nongnu.org/projects/lwip/

Installation and Usage

Tortilla's source code and binary builds can be downloaded for free from http://www.crowdstrike.com/community-tools

It supports 32-bit and 64-bit versions of the following host operating systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows 8

Tortilla ships as a single executable, Tortilla.exe. When executed, Tortilla.exe extracts several files to the user's temporary directory in order install the Tortilla Adapter and Tortilla driver if not already installed. Tortilla.exe is a 32-bit user mode program (and can thus run on 32-bit and 64-bit versions of Windows), and contains and can extract from itself both 32-bit and 64-bit driver packages for installation on 32-bit and 64-bit versions of Windows.

After Tortilla.exe extracts the appropriate driver installation package, the installation package installs or updates the device and driver as necessary and disables all of the device's network component bindings except for that of the Virtual Network Bridge. The Tortilla Client component of Tortilla.exe then establishes a secure communication channel between itself and the Tortilla driver, after which the Tortilla Client begins listening for OSI Layer 2 packets received from the VM.

Tortilla is designed with several failsafe mechanisms to allow for no-hassle ease of use. A user can run Tor before or after starting Tortilla.exe. A user can run their VM before or after starting Tortilla.exe. And a user can configure their VM platform's Virtual Network Bridge before or after starting Tortilla.exe (though the Tortilla Adapter must already be installed).

Tortilla is designed to have a minimal system footprint to make uninstallation as simple as possible. Tortilla makes no registry modifications (aside from the keys that automatically get created when installing a new device and driver), and makes no file system modifications (aside from an INI file and the installed driver package). Removal of Tortilla is as simple as deleting Tortilla.exe and Tortilla.ini, and uninstalling the Tortilla Adapter from the Network Adapters list in Device Manager.

Conclusion

Security researchers have a strong need for online anonymity. Until now, researchers were either required to use a stripped-functionality web browser as their only portal to Tor or work with an operating system used by less than 10% of the desktop market.

Tortilla is the ideal solution. It is a free and open-source project for Windows that transparently routes all TCP and DNS traffic through Tor without requiring extra hardware or an extra gateway VM.

CrowdStrike is proud to offer Tortilla to the global computer security industry in order to assist all researchers in raising the cost to the adversary.