



# Pingback: Backdoor At The End Of The ICMP Tunnel



May 04, 2021 | 7 Minute Read | by Lloyd Macrohon, Rodel Mendrez

## Introduction

In this post, we analyze a piece of malware that we encountered during a recent breach investigation. What caught our attention was how the malware achieved persistence, how it used ICMP tunneling for its backdoor communications, and how it operated with different modes to increase its chances of a successful attack. Malware using ICMP is not new but is relatively uncommon. Because of this, and the presence of certain strings, we decided to name this malware ‘Pingback’. Below we demonstrate how Pingback’s protocols work and also provide sample code on how we interacted with the malware.

We begin by looking at how Pingback achieves persistence through DLL hijacking.

## Persistence through DLL Hijacking

DLL (Dynamic Link Library) hijacking is a technique that involves using a legitimate application to preload a malicious DLL file. Attackers commonly abuse the Windows DLL Search Order and take advantage of this to load a malicious DLL file instead of the legitimate one.

The file we investigated was a DLL file called ***oci.dll***. We knew that the file was suspicious during our initial triaging, but we could not figure how it was loaded into the system because the DLL was not loaded through traditional rundll32.exe.

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

[Cookie Preferences](#)

Accept All Cookies

Decline All Cookies



Hi there! How can I help you?



File Size:	66.00 KiB (67584 bytes)
File Entropy:	5.95211
Created:	03/23/2021 16:40:49:221
Modified:	03/18/2021 15:00:54:00
Accessed:	03/23/2021 16:40:49:221
SSDEEP:	1536:b8TTUkboM8m64G95k3TdvdNs5Dpqil d52cl:b8TTLbT36x9AY5r1 d53
CRC32:	B6535337
ImpHash:	69A080EB62533E53F1ABDE958E9FB49D
MD5:	264C2EDE235DC7232D673D4748437969
SHA1:	0190495D0C3BE6C0EDBAB0D4D8D5A7E122EFBB3F
SHA256:	E50943D9F361830502DCFDB00971CBEE76877AA73665245427D817047523667F
Authentihash(PE256):	31BCF95DD6AC543D640030DE7D91A9C66A7D785EA7DCA0F3C3E768D0164F3A85
ProductVersion:	10, 2, 0, 1
FileVersion:	10, 2, 0, 1
FileDescription:	Oracle Call interface
ProductName:	Oracle Call interface
InternalName:	OCI
LegalCopyright:	Copyright (C) 2010

Figure 1: oci.dll file information

We found out later that it got loaded through a legitimate service called **msdtc** (a.k.a Microsoft Distributed Transaction Coordinator). This service, as the name suggests, coordinates transactions that span multiple machines, such as databases, message queues, and file systems.

It turns out the **msdtc** service indirectly loads *oci.dll* through MSDTCTM.DLL that loads an ODBC library to support Oracle databases called MTXOCI.DLL. This library searches for and tries to load three Oracle ODBC DLLs which include *oci.dll*, *Sqllib80.dll*, and *xa80.dll*.

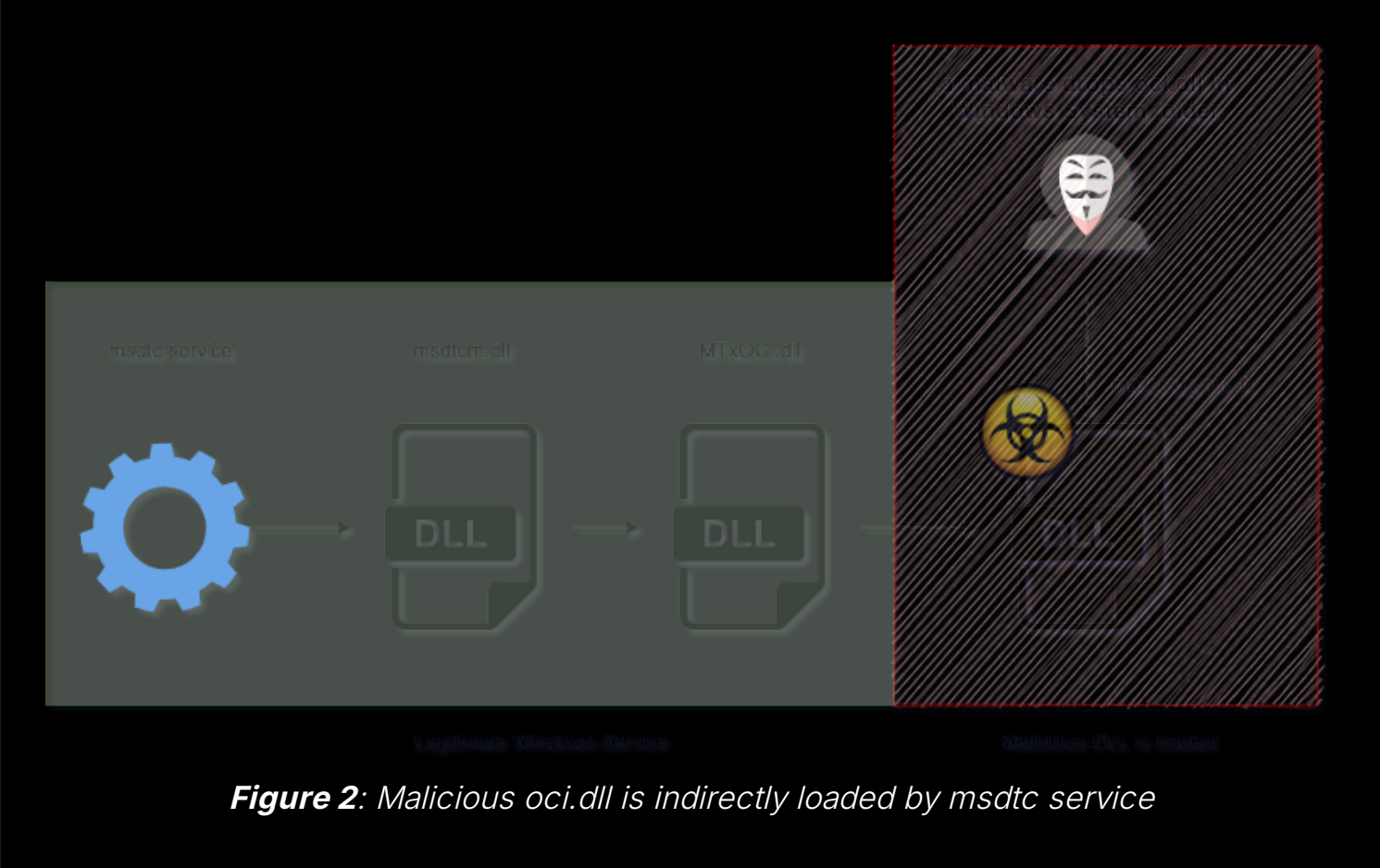


Figure 2: Malicious oci.dll is indirectly loaded by msdtc service

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

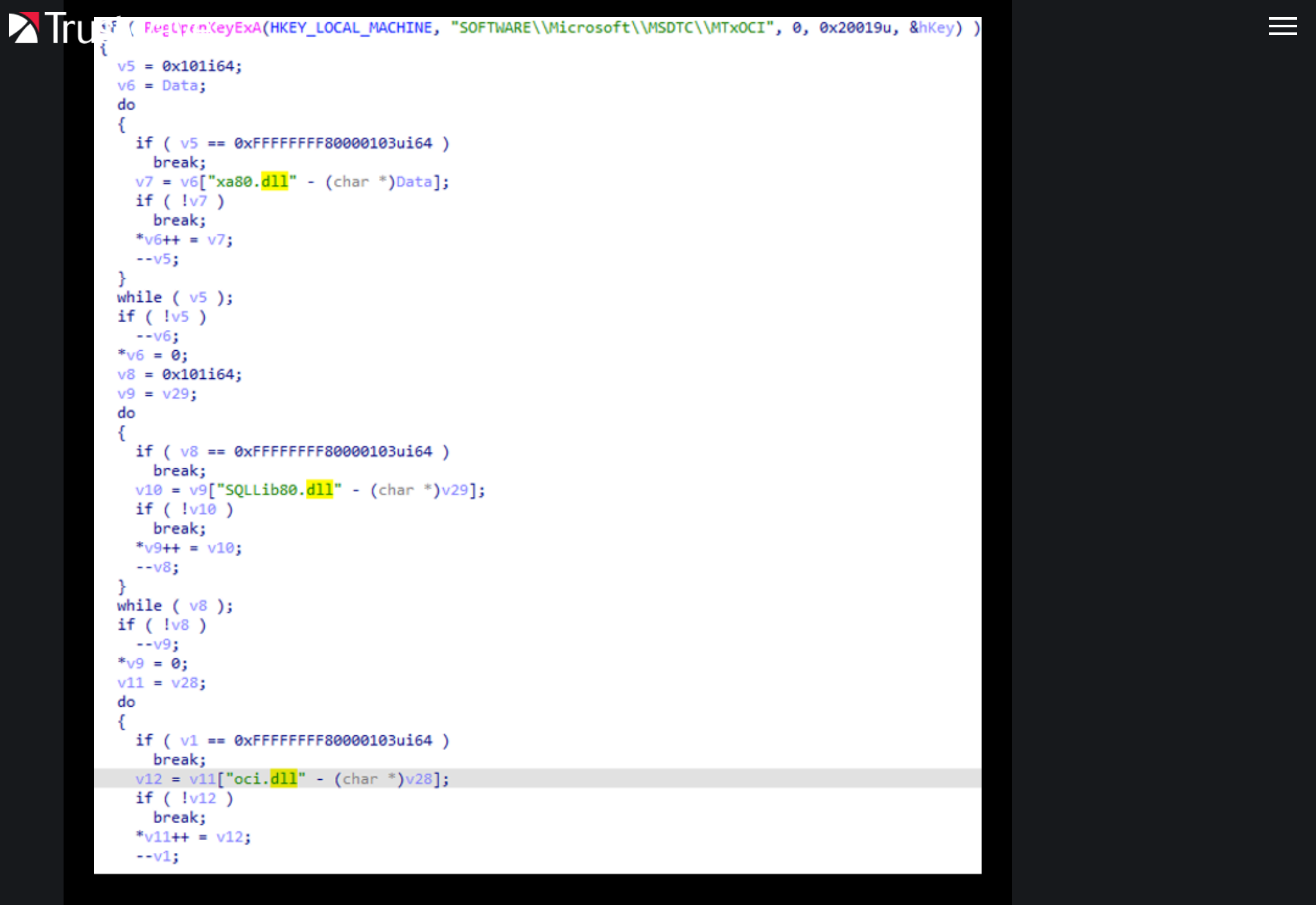
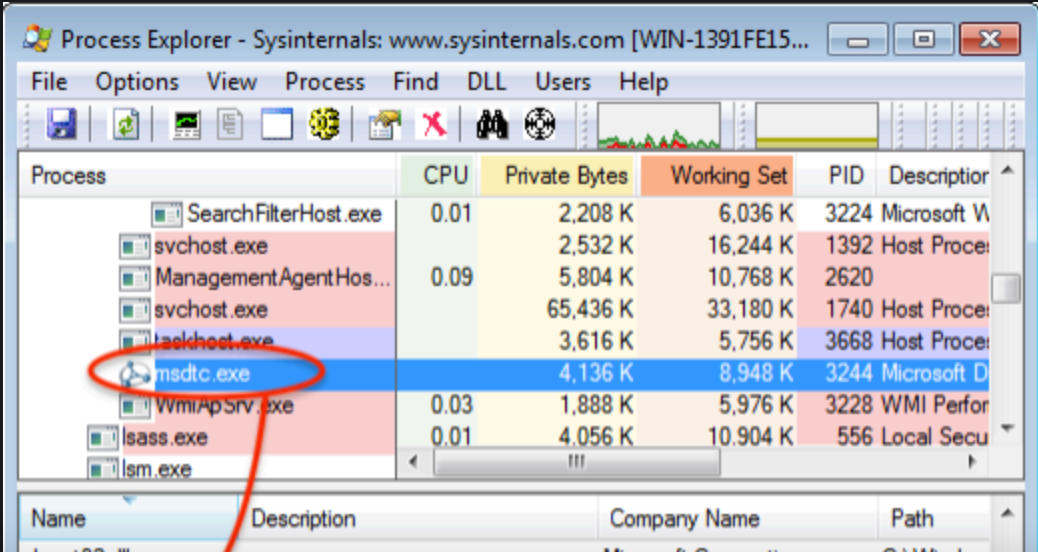



Figure 3: MTxOCI.DLL loads three plugin DLLs that support the Oracle ODBC interface

By default, the three Oracle DLLs do not exist in the Windows system directory. So, in theory, an attacker with system privileges can drop a malicious DLL and save it using one of the DLL filenames that `MTxOCI` loads. We have experimented with dropping all three DLL filenames but only **oci.dll** was successfully loaded by the service.



When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

Trustwave

**msdtc** by default does not run during start-up. To remain persistent, the **msdtc** service needs to be configured to start automatically, so the attacker would need system privileges to reconfigure the **msdtc** startup type. It can be done manually using SC command, via malicious scripts, or through a malware installer.

Our theory is that a separate executable installed this malware. In fact, after a bit of hunting, we [found](#) a sample in VirusTotal with similar IOCs that installs **oci.dll** into the Windows System directory and then sets **msdtc** service to start automatically.

```
if ( v5 == -1 || (v5 & 0x10) == 0 )
{
    WinExec("sc config msdtc start= auto", 0);
    Sleep(0x1F4u);
    WinExec(
        "reg add HKLM\\SYSTEM\\CurrentControlSet\\Services\\msdtc /v objectname /t REG_SZ /d \\\"LocalSystem\\\" /f",
        0);
}
else
{
    WinExec("sc config msdtc obj= Localsystem start= auto", 0);
}
Sleep(500u);
WinExec("sc start msdtc", 0);
```

Figure 5: A loader configuring **msdtc** service to start automatically

We also observed during our analysis that in a VMware environment, the VM Tools service also loads MTXOCI and eventually loads the malicious OCI.DLL.

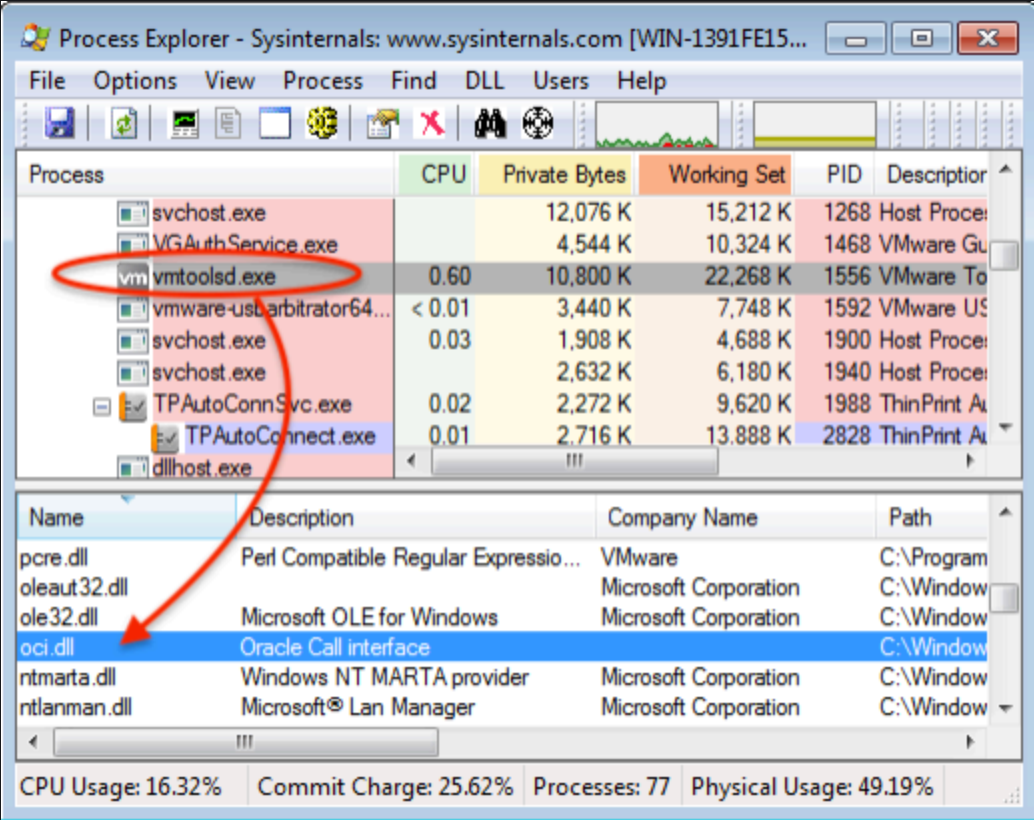


Figure 6: In Process Explorer, we found that OCI.DLL is also loaded by VMTools service in a VMware Environment

So that is the DLL loading part. But before turning our attention to Pingback itself and its operation, let us first lay out what is ICMP and how ICMP tunneling works.

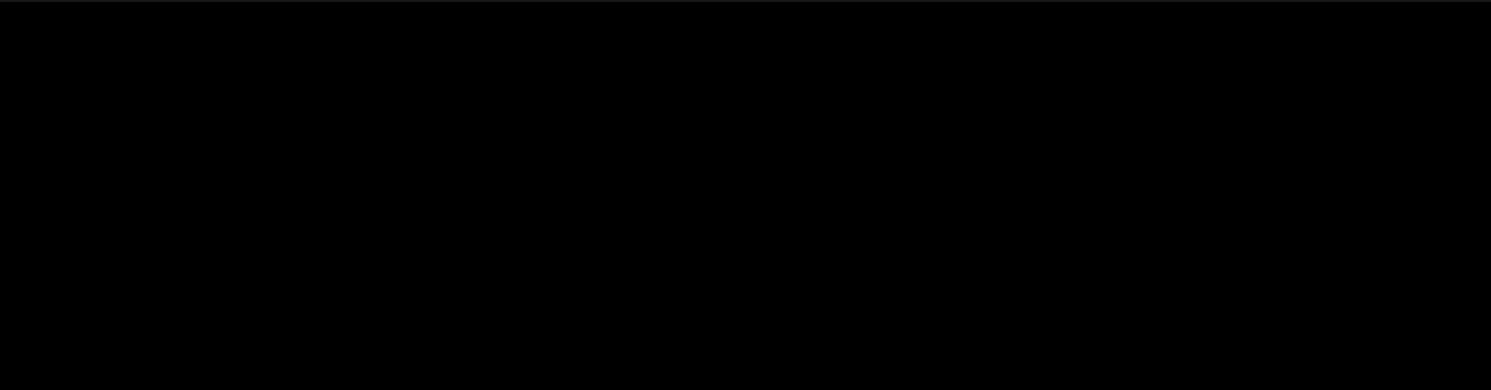
## ICMP Foundation

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

its main communication. This has the effect of being hidden from the user as ports cannot be



To explain ICMP tunneling, let us first understand an ICMP packet:



**Figure 7:** A diagram of an ICMP packet. ICMP data size varies, we assume that the IP maximum transmission unit is 1500 bytes. The packet size limit for an ICMP Data is maximum allowed size of an IPv4 network packet, minus the 20 byte IP header and 8 byte ICMP header.

An ICMP packet is built on top of the IP layer and has an 8 byte ICMP header. The packet size limit for ICMP data is a maximum allowed size of an IPv4 network packet, minus the 20 byte IP header and 8 byte ICMP header. Or approximately 64K. The ICMP data is determined by the message type. The message types are defined here: <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>

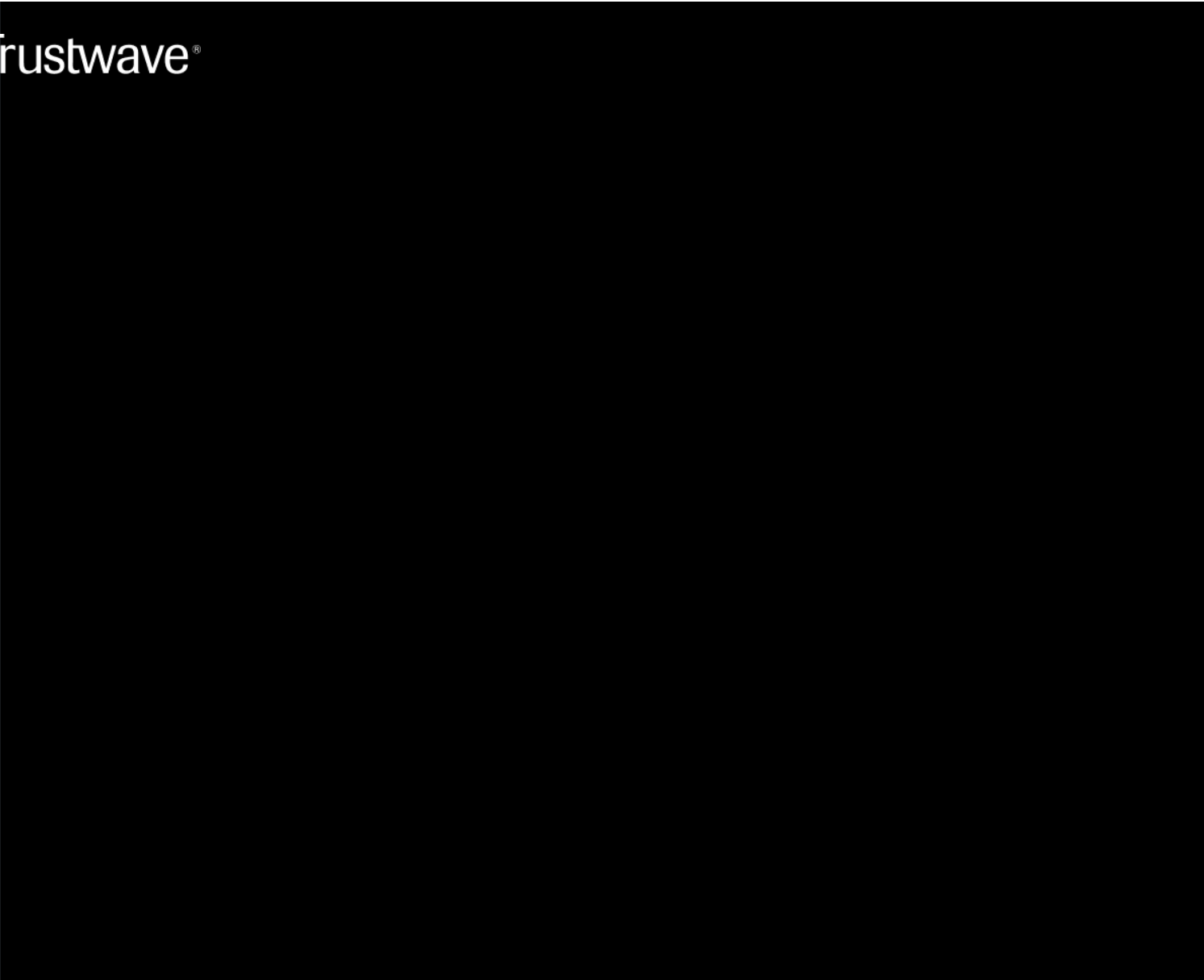
An ICMP tunnel mainly uses these two types:

Code	Type	Description
0	Echo Reply	ping reply
8	Echo	ping

In the diagram below, A echo packet header defines the ICMP type, code, checksum, identifier and sequence number. And lastly, the ICMP data section is where an attacker can piggyback an arbitrary data to be sent to a remote host. The remote host replies in the same manner, by piggyback an answer into another ICMP packet and sending it back.

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

**Figure 8:** Echo packet. The size of the ICMP data sent by the attacker is always 700 bytes.



**Figure 9:** Packet capture of the attacker's ICMP packet

## ICMP Tunneling

Now that we have laid out the foundation and how an attacker can piggyback data on the ICMP packets, we are ready to explain exactly how exactly this malware works.

Pingback specifically uses the echo (ping) request or type 8 ICMP message. It starts a sniffer for every IP address on the host, spawning a thread to sniff packets on each individual IP address. To distinguish between its own packets and other packets, the sniffer ignores anything else that's not an ICMP echo packet and does not contain the ICMP sequence number 1234, 1235 or 1236. It also ignores packets not destined for the specified IP address.

It then interprets the data in the following format:

**Figure 10:** Malware's ICMP data is represented by this C structure. See appendix below for detailed information of the **cmd** and **cmd\_line** fields.

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

- **shell** – execute a shell
- **download** – 3 different modes of download are provided:
  - Mode 1: Infected host connects back to the attacker’s host (works well if incoming TCP connections are blocked by firewall)
  - Mode 2: Infected host opens a socket on a specified port and waits for the attacker to connect.
  - Mode 3: Purely ICMP-based, but this is very slow and the current implementation is not very reliable in terms of flow control.
- **upload** – also supports 3 different modes, similar to the Download command.
- **exec** – execute a command on the infected host.

This is interesting, you can see Pingback uses a combination of ICMP for initiating any of the commands and TCP for better performance and reliability. A pure ICMP mode is also provided but is not very reliable.

## Protocol example

To download a file in mode 1, the attacker performs:

- Create a socket and listen on a port
- Send the following ICMP packet:
  - ICMP echo request with sequence 1234
  - Payload contains: “download”, name of file to download and, IP address and port of the newly created socket, this is where the malware will stream the file to
- Wait for connections and receive data. There is an additional protocol here which includes is reasonably simple (send and receive) and wait for “END\x00” string.

We have provided a source to demonstrate all three modes and most of the commands supported by Pingback.

Source available here:

<https://github.com/SpiderLabs/pingback>

We have also prepared a video to demonstrate how our client interacts with the malware running in an isolated infected system.

## Final Words

ICMP tunneling is not new, but this particular sample piqued our interest as a real-world example of malware using this technique to evade detection. ICMP is useful for diagnostics and performance of IP connections in the real world. It is very useful to have them enabled

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

# Appendix

**cmd** – bot commands and may be any of the following:

- *exep* (execute process) – execute a binary/command on the remote host
- *download* (download mode 1) - attacker’s initial connection is done via ICMP and appears as a ping packet. The ICMP echo packet contains data that specifies the attacker’s host and port to where the malware connects back. The ICMP data also contains a file path that the attacker requests. Using the host and port information, the malware creates a new socket, then transmits the requested file back to the attacker.
- *upload* (upload mode 1) – attacker’s initial connection is done via ICMP. The malware receives the initial connection then connects back to attacker’s host and port specified in the ICMP Echo packet. It then receives the file from the attacker to be saved in the infected system’s local disk
- *download2* – (download mode 2), initial ICMP packet is sent by the attacker. The ICMP echo packet contains the requested filename and path in the infected system. It also contains a port number where the malware will bind and listen to. The malware then waits for the attacker to connect, afterward, it begins transmitting the requested file.
- *upload2* – (upload mode 2), initial ICMP packet is sent by the attacker. The ICMP echo packet contains the filename of the file to be received. It also contains a port number where the malware will bind and listen to. The malware then creates the file in the remote host and waits for the attacker to connect. Once connection is established, the attacker begins transmitting the file content to the remote host
- *download3* – (download mode 3), a file is sent to the attacker purely through ICMP data. Although this mode is more covert as it appears as ping packets only, this is slower than using TCP directly as only 1 packet can be transmitted at a time. The malware has to wait for acknowledgment from the attacker’s end.
- *upload3* – (upload mode 3), same as download mode 3 – although the attacker uploads the file purely through ICMP. Also slower and unreliable but more covert than other modes.
- *shell* – request malware to connect back to the attacker with a shell. Initial request is done via ICMP packet containing information including attacker’s host IP and port to where the malware makes a TCP connection.

**cmd\_line** - In *exep* command, this variable holds the command to be executed on the remote host. While in *download* and *upload* command, this variable contains the remote file name.

# IOC:

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

Destination:



ICMP Type: 8  
Sequence Number: 1234|1235|1236  
Data size: 788 bytes







PCAP: <https://github.com/SpiderLabs/IOCs-IDPS/tree/master/Pingback>

Yara:

```
rule PingBack
{
    meta:
        description = "This rule detects PingBack malware"
        author = "Trustwave SpiderLabs"
        date = "May 4th, 2021"

    strings:
        $string1 = "Sniffer ok!" ascii
        $string2 = "lock2" ascii
        $string3 = "recvfrom failed" ascii
        $string4 = "rexec" ascii
        $string5 = "exep" ascii
        $string6= "download" ascii
        $string7 = "download2" ascii
        $string8 = "download3" ascii
        $string9 = "upload" ascii
        $string10 = "upload2" ascii
        $string11 = "upload3" ascii
        $string12 = "cmd.exe" ascii
        $string13 = "PingBackService" ascii

    condition:
        all of them
}
```

Share:    

Stay Informed

Sign up to receive the latest security news and trends straight to your inbox from Trustwave.

Business Email\*

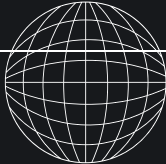
Subscribe

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)


ABOUT TRUSTWAVE

Trustwave is a globally recognized cybersecurity leader that reduces cyber risk and fortifies organizations against disruptive and damaging cyber threats. Our comprehensive offensive and defensive cybersecurity portfolio detects what others cannot, responds with greater speed and effectiveness, optimizes client investment, and improves security resilience. Learn more [about us](#).

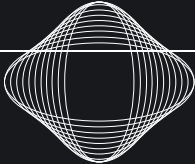
Latest Intelligence



2024 Trustwave Risk Radar Report: Cyber Threats to the Retail Sector →



Hooked by the Call: A Deep Dive into The Tricks Used in Callback Phishing Emails →



How Threat Actors Conduct Election Interference Operations: An Overview →

Related Offerings

- Penetration Testing
- Digital Forensics & Incident Response
- Threat Intelligence as a Service
- Threat Hunting

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)

Request a Demo



# Stay Informed

Sign up to receive the latest security news and trends straight to your inbox from Trustwave.

Business Email\*

Subscribe

Leadership Team

Our History

News Releases

Media Coverage

Careers

Global Locations

Awards & Accolades

Trials & Evaluations

Contact

Support

Security Advisories

Software Updates



Legal

Terms of Use

Privacy Policy

Copyright © 2024 Trustwave Holdings, Inc. All rights reserved.

When you visit our website, we and our third-party service providers may use cookies and similar technologies to collect information about you. This information may be used to measure website usage and performance, optimize user experience, and provide targeted advertisements. For more information on our use of cookies and tracking technologies please review our [Privacy Policy](#)