NTA Monitor UDP Backoff Pattern Fingerprinting White Paper

Author: *Roy Hills*
Date: *17 January 2003*

Table of contents

1. Abstract **1**

2. Overview **1**

3. Issues **2**

4. Example program **3**
Program Overview 3
IKE Packet exchange diagram 4
Example output 5

5. Contact address **6**

6. Glossary of terms **6**

1. Abstract

This paper discusses how it is possible to determine which implementation of a UDP service is being used from the retransmission backoff pattern. It uses IKE (Internet Key Exchange) as an example UDP service which can be identified in this way, although the technique may also be applicable to other UDP services.

The paper also describes an example program called "ike-scan" which is able to discover and identify IPsec VPN systems running IKE. This program is publicly available under the GNU Public License (GPL).
See http://www.nta-monitor.com/ike-scan/ for details.

2. Overview

Although most services on the Internet use the TCP transport, some use UDP instead. Because UDP is not a reliable transport, it is up to the application to provide the reliability itself if needed. The main technique used to ensure reliability is retransmission with backoff which allows the application to tolerate lost or damaged packets.

Retransmission backoff involves re-sending a packet if a response is not received from the peer within a given time, the assumption being that the original packet must have been lost or damaged in transit. There are several variables involved with the retransmission strategy, including:

a) How long to wait before re-sending the packet?
b) How should the re-send delay change for subsequent packets, e.g. should the delay remain constant, or should it increase with each re-send. If the delay is to increase, what algorithm should be used (e.g. linear or exponential) and what parameters should be used?
c) How many packets should be sent before giving up?

Often, the exact retransmission strategy is not specified by the relevant standard which means that each developer will typically choose their own scheme. Because there are a number of variables involved, and there is no “obviously correct” choice of strategy, this results in most implementations having distinct backoff patterns or “fingerprints”. This distinctive fingerprint could be used to determine which vendor’s implementation is being used.

Potentially any UDP based service which needs to implement reliable data transfer and does not have the retransmission strategy defined by the appropriate standard may be subject to identification through this backoff fingerprinting method. The specific UDP service which has been investigated in detail is IKE.

For IKE, the use of retransmission is mandated by RFC 2408 (ISAKMP - the protocol framework used by IKE) Section 5.1 but the exact backoff strategy is not defined. RFC 2408 suggests basing the retransmission times on measured round-

trip times. However this is not practical for the first transmitted packet because there are no previous round-trip times to use. It is first packets transmitted by the VPN server which are used by the "ike-scan" program to determine the backoff fingerprint.

The retransmission and backoff strategy for various different IKE implementations has been studied, and it has been found that:

a) Most, if not all, IPsec VPN vendor implementations have different IKE retransmission and backoff strategies; and

b) It is possible to reliably match these patterns to determine which IKE implementation a particular host is using.

Sometimes the backoff pattern changes from one version of a product to another which provides more information and allows different versions of a particular implementation to be distinguished from each other.

3. Issues

Although just being able to discover an IPsec VPN system running IKE and determine which IKE implementation it is using is not a vulnerability in itself, this information can be valuable to a potential attacker.

For example, knowing that there is an XYZ brand of VPN server at a given address could prompt an attacker to download the appropriate VPN client and try some username/password guessing. Alternatively, the attacker could search for known vulnerabilities associated with the XYZ VPN server.

Some IKE implementations don't log IKE activity if the handshake does not complete. Because it is not necessary to complete the IKE handshake to discover and fingerprint the system, these systems will not log the scanning activity and their owners will not be aware that their system has been scanned. "ike-scan" does not complete the IKE handshake and can therefore be used to check if a given implementation will log this sort of scanning.

This choice not to log if the handshake doesn't complete, and particularly if the initial cookie exchange does not complete, may be based on the recommendation in RFC 2408 section 1.7.1 that "A 'cookie' ... is aimed at protecting the computing resources from attack without spending excessive CPU resources ..." i.e. nothing "expensive" should be done until the cookie exchange completes. Although this is normally taken to apply to the CPU-intensive cryptographic functions, it could also be applied to logging if log storage were considered expensive. On the other hand, RFC 2408 section 5.1 says that "If the retry counter reaches zero (0), the event, RETRY LIMIT REACHED, MAY be logged in the appropriate system audit file". This would also apply when a system is scanned. In summary, the RFC seems to leave the choice to log or not up to the implementation.

In the course of performing many security assessments and penetration tests for customers, we have found that VPN systems often provide full access to the internal network which makes them tempting targets to an attacker. In addition, many people assume that their VPN servers are invisible and impenetrable which is a dangerous assumption given that the "ike-scan" program shows that IPsec VPN systems can be discovered and identified. When this potential for discovery and identification is combined with the fact that several VPN vulnerabilities have been discovered in the

past few months, it would seem to be only a matter of time before hackers start to target VPN systems.

The aim of this white paper and the associated "ike-scan" program is to demonstrate the problem so that it is understood by the security community and VPN vendors. The program also allows organisations to test their own networks to see what information a hacker could discover and close any holes before they can be exploited.

4. Example program

Program Overview

The program "ike-scan" demonstrates detection and identification of IPsec VPN systems. The backoff patterns are stored in a text file which makes it easy to add new patterns as they are discovered.

This program is available for free download from:
http://www.nta-monitor.com/ike-scan/

ike-scan sends an initial IKE main-mode packet to each of the specified hosts and records all the responses returned. It will display the responses received which discovers which hosts are running IKE and will return a response (most IKE implementations will respond in the default configuration, but not all). It can also record and display the retransmission backoff pattern for each responding host and attempt to match this pattern against a database of known patterns to "fingerprint" the IKE implementation.

The program handles retry and retransmission with backoff to cope with packet loss. It also limits the amount of bandwidth used by the outbound IKE packets.

IKE Packet exchange diagram

The IKE packet exchange between *ike-scan* and a VPN server which returns a handshake response is shown in the diagram on the next page.

In this packet exchange, the *ike-scan* program sends a packet containing a Header and an SA (Security Association) to the VPN server. The VPN server then responds with a Header and SA. *ike-scan* records the time of this response packet for later fingerprinting, but it does not respond to it. Because the VPN server does not receive a response from ike-scan, it assumes that the packet must have been lost, so it re-sends the packet after a delay. As the VPN server never received any response from ike-scan, it keeps resending the same packet using its retransmission strategy until it gives up.

When the last packet has been received from the VPN server, *ike-scan* has the receive times for all of the packets. These receive times can be used to display the retransmission strategy and also attempt to match this strategy against known strategies.

Key

1), 2), etc. are packet numbers

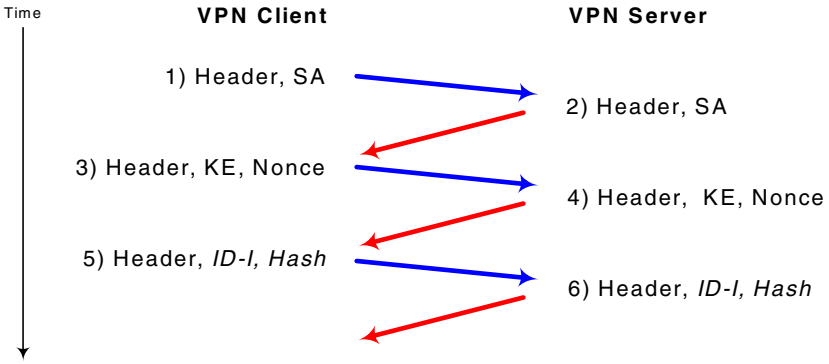
Hash means packets are encrypted from this point on

KE = Key Exchange
SA = Security Association
ID-I = Identification of the Initiator (VPN Client)

IKE-Scan Diagram

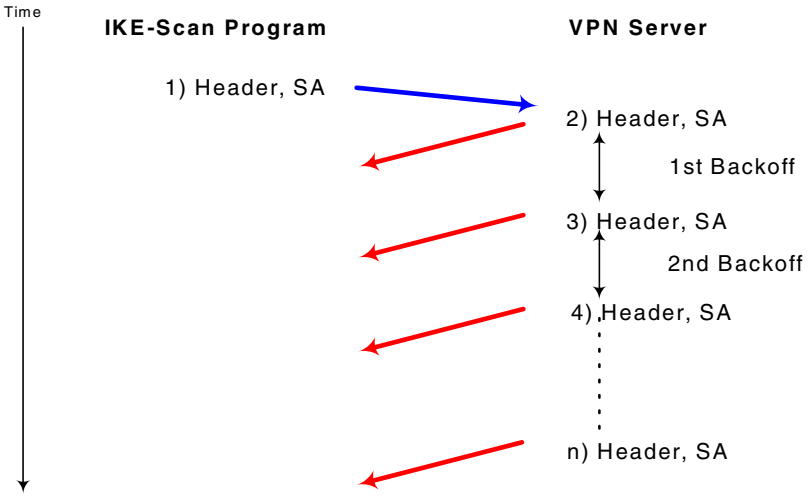
Normal Main Mode Exchange

The diagram below shows the normal packet exchange between a VPN Client and a VPN Server required to authenticate and set up the first phase of an encrypted channel.



IKE-Scan Main Mode Exchange

The diagram below shows that the IKE-Scan Program does not respond to packet 2 from the VPN Server, as per the diagram above. Therefore the VPN Server continues to re-send the same packet (awaiting a reply) using a pre-determined "backoff strategy" to set the frequency and number of re-transmission attempts, before timing out the connection.



Example output

An example run of the program is shown below:

$ *ike-scan* --showbackoff 172.16.2.2 10.0.1.98
Starting *ike-scan* v1.0 with 3 hosts (http://www.nta-monitor.com/ike-scan/)
172.16.2.2 IKE Handshake returned (1 transforms)
10.0.1.98 IKE Handshake returned (1 transforms)

IKE Backoff Patterns:

IP Address	No.	Recv time	Delta Time
172.16.2.2	1	1042797936.905288	0.000000
172.16.2.2	2	1042797938.901378	1.996090
172.16.2.2	3	1042797940.904158	2.002780
172.16.2.2	4	1042797942.906987	2.002829
172.16.2.2	5	1042797944.909644	2.002657
172.16.2.2	6	1042797946.912480	2.002836
172.16.2.2	7	1042797948.915286	2.002806
172.16.2.2	8	1042797952.920635	4.005349
172.16.2.2	9	1042797956.926155	4.005520
172.16.2.2	10	1042797960.931677	4.005522
172.16.2.2	11	1042797964.937201	4.005524
172.16.2.2	12	1042797968.942691	4.005490
172.16.2.2	Implementation guess: Firewall-1 4.1/NG		
10.0.1.98	1	1042797937.070152	0.000000
10.0.1.98	2	1042797952.061102	14.990950
10.0.1.98	3	1042797967.064137	15.003035
10.0.1.98	Implementation guess: Cisco IOS / PIX		

In the above example, the *ike-scan* program was run with the –showbackoff option against the two hosts 172.16.2.2 and 10.0.1.98. The program first discovers that both hosts are running IKE and that both of them will return an IKE handshake response as shown by the "IKE Handshake returned". lines.

The program then records and displays the retransmission backoff pattern that the VPN servers use when re-sending its response to the IKE packet sent by ike-scan. The pattern responses contain the following four columns:

IP Address The IP address of the VPN server that this pattern relates to.

No. The number of the response packet from this host with the first response packet being 1.

Recv time The time when this response packet was received. This time is shown as the number of seconds and microseconds since midnight on Jan 1, 1970 (the Epoch used by Unix systems).

Delta Time The difference between the time when this response packet was received and the time when the previous response packet was received. This is always zero for the first response packet. The difference is shown in seconds and microseconds.

5. Contact address

Please send any questions or comments to: ike-scan@nta-monitor.com

See the *ike-scan* homepage at: http://www.nta-monitor.com/ike-scan/

By Post:

NTA Monitor Limited
14 Ashford House
Medway City Estate
Rochester
Kent ME2 4FA
UK

6. Glossary of terms

IKE Internet Key Exchange. The protocol used by IPsec to exchange keys and authenticate the users or devices at either end of the VPN. IKE is defined in RFC 2409.

IPsec Internet Protocol SECurity, security functions (authentication and encryption) implemented at the IP level of the protocol stack. Most VPN implementations use IPsec.

TCP Transmission Control Protocol. The most common transport protocol in the TCP/IP protocol suite. TCP is a reliable protocol. TCP is defined in RFC 761.

UDP User Datagram Protocol. One of the transport protocols in the TCP/IP protocol suite. UDP is an unreliable protocol, that is UDP does not guarantee data delivery. UDP is defined in RFC 768.

VPN Virtual Private Network. Allows local area networks to communicate across public networks such as the Internet, typically over an encrypted channel.

RFC Request for Comments. The standards documents for Internet protocols.

RFCs are available from http://www.ietf.org/rfc.

The RFCs relating to IKE are:

RFC 2407 - The Internet IP Security Domain of Interpretation for ISAKMP

RFC 2408 - Internet Security Association and Key Management Protocol

RFC 2409 - The Internet Key Exchange (IKE)

Cookie A unique 64-bit value used by IKE to identify peers and prevent some Denial of service attacks.