What is GRE?

John P. Harris Jr.

Assignment: GSEC Version 1.2d

Introduction

In many journeys through the world of information security, IT folks are barraged with many acronyms. These become a blur of letters and few new IT folks take the time to learn more than just the description of the acronym. Recent travels by the author through a Cisco Site-to-Site VPN implementation have reveled many of these acronyms. The first was GRE (Generic Routing Encapsulation), which was not defined or explained in the Cisco documentation. It is better known to today's IT folks as IP Protocol 47. Is this really what GRE is? What does GRE do? What is GRE used for? These questions and more should be answered in the following document. This document is intended to be a brief overview of GRE. There will be references to more in-depth documentation throughout the paper.

In the Beginning

A brief definition of GRE follows:

> "(GRE) A protocol which allows an arbitrary network protocol A to be transmitted over any other arbitrary network protocol B, by encapsulating the packets of A within GRE packets, which in turn are contained within packets of B." [1]

By definition, GRE is more than just IP protocol 47! GRE was originally envisioned as a way to encapsulate any protocol in any other protocol as described in the original RFC (Request For Comments), RFC1701. This encapsulation would take place at Layer 3 of the OSI model, taking the form of a delivery header followed by a GRE header followed by a Payload Packet as shown in Figure 1. [2] Typically this is a 24 bit GRE header.

```
 -----------------------------------
|                                   |
|          Delivery Header          |
|                                   |
 -----------------------------------
|                                   |
|            GRE Header             |
|                                   |
 -----------------------------------
|                                   |
|          Payload Packet           |
|                                   |
 -----------------------------------
```

Figure 1: Original GRE Encapsulated Packet.

To elaborate more on this, say a system needs to send some information to another system. To accomplish this seems pretty elementary, but if you add in the variable that the network somewhere between the two systems is running a protocol that is not what the two systems want to speak to each other with, therein lies the problem. The system would then have to encapsulate its information into the protocol that could be routed and then sent over to the other system that would unwrap the data. This could also be performed by other systems (like routers) in between the two systems that need to talk. These other devices will usually be positioned on the edges of the networks that are speaking the different protocol (i.e. the Internet). This is what GRE was designed to do.

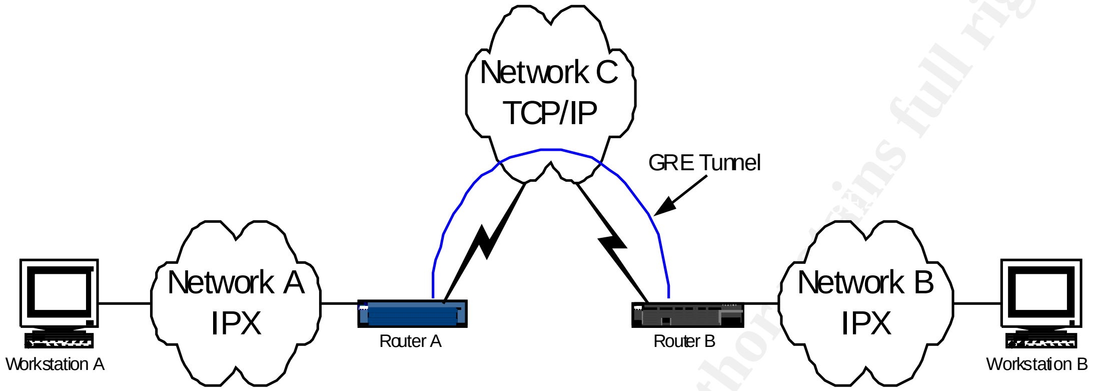

Figure 2: Example GRE Architecture

For a more real life example, say you had two machines (Workstation A and Workstation B) on different networks that wanted to exchange Novell IPX information (Figure 2). The limitation is that the networks are separated by a TCP/IP WAN (Wide Area Network) link. This is the perfect situation for GRE. The routers (Router A and Router B) on either end of the WAN link would be set up for a GRE tunnel with each other to pass the IPX information encapsulated inside TCP/IP packets. The flow would be as follows:

1. Workstation A sends a packet out onto Network A that needs delivered to Workstation B on Network B.
2. Router A takes the packet and examines it. It knows the packet type and the fact that it needs to go to Router B so it can be delivered as requested.
3. Router A then encapsulates the packet into GRE using TCP/IP.
4. The new GRE packet is sent via Network C to Router B.
5. Router B receives the packet, identifies it as a GRE packet.
6. Router B "unwraps" the packet.
7. Router B then forwards the packet out onto Network B to be delivered to Workstation B.
8. Workstation B receives the packet and processes it.

This is really an oversimplified example since there are numerous occasions where the packet may exceed size limit requirements and has to be broken apart and then re-assembled by the routers at each end. Some applications may even fail because of this. Another concern is the original packet's TTL must be decremented (so the packet does not live forever!).

What's the Pretty Wrapper?

To take a more detailed look, the packet header on a GRE packet is shown:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-----------------------------------------------------------------
|C |          Reserved0             |  Ver  |          Protocol Type            |
-----------------------------------------------------------------
|          Checksum (optional)            |            Reserved1 (optional)         |
-----------------------------------------------------------------
```

The Checksum bit (C) is the first bit and it determines if there is valid data in the Reserved1 field and data in the Checksum field. Since the Reserved1 field is for future use, it must always be set to zero if present. The Checksum field contains the IP checksum sum of all the 16 bit words in the GRE header and the payload packet. [3]

Bits 1-5 should be zero unless the systems have implemented the original RFC1701. If this is not the case, the packet should be discarded as damaged. Bits 6-12 are reserved and should always be zero. As well, bits 13-15 (Ver) should also be zero.

The next two octets (Protocol Type) contain information about the protocol type of the payload packet. These are defined in RFC1700. If these octets have no match for the protocols listed in RFC1700, the whole packet should again be discarded.

If the routing field were present, it would contain a list of SREs (Source Route Entries). Each of these would take the following form:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-----------------------------------------------------------------
|        Address Family                  |    SRE Offset    |    SRE Length        |
-----------------------------------------------------------------
|                                   Routing Information                                   *
-----------------------------------------------------------------
```

* The Routing Information field will terminate with a null SRE.

The first field, Address Family, is a two-octet value that explains the setup of the Routing Information Field. The Routing Information Field contains the data to assist in routing the packet. The SRE Offset and SRE Length fields contain data that set the spacing in the Routing Information Field and the whole SRE, respectively. In some cases, the routing information from the original packet may be transferred into the GRE header. The example above does not display this behavior.

How this Applies Today

It didn’t take the technical folks long to see that this encapsulation could be very useful and to actually start to suggest these possible uses. Following RFC1701 was RFC1702, which suggested encapsulating IP packets in IP packets. This RFC was actually released as a companion piece for RFC1701. It is in this case where the encapsulated packets’ TTL, TOS, and IP security options can be copied directly into the delivery packet. For Ipv4, the IP protocol type will be set to 0x800. [4] This is also where IP protocol 47 will be used for GRE packets.

VPN technology seems to be the largest current exploration into GRE. What would any topic of discussion be without an example of how the largest OS manufacturer uses the technology? Of course this refers to Microsoft. The big company from Redmond is using PPTP packets, given an additional PPP header, and then wrapped in GRE for its own VPN implementations. [5] Being that this is Microsoft, they of course have made some changes to the way the original RFCs implement GRE and have labeled this as “Enhanced” GRE. This does fit with the implementation proposed in RFC2637 for PPTP. [6] Examination of the implementation displays the following header:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-----------------------------------------------------------------
|C |        Reserved0          | Ver |        Protocol Type        |
-----------------------------------------------------------------
|        Key (HW) Payload Length     |        Key (LW) Call ID      |
-----------------------------------------------------------------
|                     Sequence Number (Optional)                    |
-----------------------------------------------------------------
|                  Acknowledge Number (Optional)                    |
-----------------------------------------------------------------
```

The Key (HW) Payload Length is the size of the payload excluding the GRE header.

The Key (LW) Call ID is the Peer’s Call ID for the session.

The sequence Number is the standard implementation.

The Acknowledge Number contains the sequence number of the highest numbered GRE packet received by the sending peer for this session.

Cisco implements GRE as per RFCs 1701, 1702, and 2784. Not surprisingly, Cisco provides little detail in their configuration guide for the 7100 series router [7] on the specifics of the GRE implementation. Cisco does however use GRE as the base tunnel type in its Router-to-Router Site-to-Site VPN configuration. The documentation does however mention it is a "secure" way to transport data. This statement is suspect and requires further investigation to see if the data being encapsulated is actually secure. This is beyond the scope of this document.

Security

What document would be complete without examining some of the security issues surrounding the topic being discussed? GRE, like every other topic, has security issues that need to be discussed so the limitations and benefits of any implementation can be realized.

The security of the data passed across the tunnel is left to the underlying encapsulated data. This means that the data has to be encrypted before it is encapsulated in GRE. By no means is GRE a secure method to transport data. If a vendor states that the data is secure, they should specify how the data is protected. GRE is just a tunneling protocol. For example, an attacker could invade the GRE stream and inject data that would allow them to attack systems inside the networks being tunneled. Unless the payload has been cryptographically protected, any attacker can easily capture the GRE packets and read the data being transported. This allows the attacker to gain the knowledge required to attack the systems passing the data and even other systems inside the private networks. There is no question that all private data traversing untrusted networks should be encrypted.

If the GRE tunnels are set up in such a way that the routing is done dynamically, intruders could inject routes into the network and disrupt traffic. Another extreme in this example is the fact that the intruders could actually add themselves as a GRE endpoint and have full access to not just the data being transmitted, but also the systems on the networks themselves. To defeat this, it is recommended to only use static routing across the tunnels and to leave the setup and configuration as a manual process. Another suggestion to defeat this type of attack is to have the data pass through a firewall after the GRE header is removed. Private network numbers for the tunnel interfaces that are not routed on either side of the network can also help.

The largest problem with GRE is the MTU size issue. How does this relate to security? Can you say DOS (Denial of Service)? IT professionals think of DOS attacks as being the result of crackers or hackers trying to disrupt the data flow. DOS can also be thought of from the perspective of a bad implementation! How many times has a corporation implemented some new technology without being fully aware of the limitations of the technology, only to inflict a DOS attack on themselves, costing the company money in lost time or even system down time? The MTU size limit of tunnels can do this. The majority of systems installed today leave the default MTU value set to 1500 and then eventually generate packets at this MTU size limit. The tunneling system will then try to add the 24 bit GRE header on the packet and this makes the packet too large for the MTU size. The tunneling systems begin to fragment the packets so they will fit under the MTU limit. Where this becomes an issue is when the DF (do not fragment) bit is set and the tunneling systems fragment the packet. This can cause problems for some applications that cannot process the packets once they have been fragmented. This also becomes an issue when a vendor such as Cisco wants to send an ICMP packet out to have the sending system retransmit the packets at a smaller size, but ICMP is blocked on the transmitting system (such as a web server that only responds to TCP port 80). [8]

Another source of problems is the fact that in today's environment, companies are looking to save costs on all levels, which in turn is forcing a lot of site-to-site traffic across the public Internet. Most companies' executives do not understand that there is no single body responsible for servicing problems across the Internet. This can also result in a self inflicted DOS problems. It also introduces systems exposed to the edge of the Internet that are reachable by attackers.

Conclusion

In conclusion, it is easy to now see that GRE is a method of data encapsulation that can be used over any protocol. This allows data to be transported across networks that are running different protocols than the originating network. In the IP protocol, GRE is better known as IP protocol 47. In current situations, GRE is used for tunneling different varieties of VPN traffic such as Microsoft's PPTP and Cisco's Site-to-Site traffic. Although this method does have a lot of advantages, it also has some drawbacks specific to each implementation. This occurs because the data integrity in the packet is left to the implementers. GRE also has some generic problems such as the problem with fragmenting packets due to the MTU size. GRE has gone through a small evolution and may continue to evolve depending upon its usefulness in tomorrow's technologies.

Cited Sources:

[1] Howe, Denis. "Free Online Dictionary of Computing." 19 July 1998. URL: http://burks.bton.ac.uk/burks/foldoc/61/46.htm 20 May 2001.

[2] Hanks, Stan, Netsmiths, Ltd., Li, Tony, Farinacci, Dino, Traina, Paul, Cisco Systems. "Generic Routing Encapsulation (GRE)." Network Working Group, Standards Track, RFC1701, October 1994. URL: http://www.ietf.org/rfc/rfc1701.txt 20 May 2001.

[3] Farinacci, Dino, Li, Tony, Procket Networks, Hanks, Stan, Enron Communications, D. Meyer, David, Cisco Systems, Traina, Paul, Juniper Networks. "Generic Routing Encapsulation (GRE)." Network Working Group, Standards Track, RFC2784, March 2000. URL: http://www.ietf.org/rfc/rfc2784.txt 20 May 2001.

[4] Hanks, Stan, Netsmiths, Ltd., Li, Tony, Farinacci, Dino, Traina, Paul, Cisco Systems. "Generic Routing Encapsulation over IPv4 Networks." Network Working Group, Standards Track, RFC1702, October 1994. URL: http://www.ietf.org/rfc/rfc1702.txt 20 May 2001.

[5] Microsoft Corporation. "VPN Tunnels – GRE Protocol 47 Packet Description and Use." Article ID: Q241251. 22 October 2000. URL: http://support.microsoft.com/directory/article.asp?id=KB;EN-US;Q241251 20 May 2001.

[6] Hamzeh, Kory, Ascend Communications, Pall, Gurdeep, Microsoft Corporation, Verthein, William, 3Com, Taarud, Jeff, Copper Mountain Networks, Little, W. Andrew, ECI Telematics, Zorn, Glen, Microsoft Corporation. "Point-to-Point Tunneling Protocol (PPTP)." Network Working Group, Standards Track, RFC2637, July 1999. URL: http://www.ietf.org/rfc/rfc2637.txt 20 May 2001.

[7] Cisco Systems. "Cisco 7100 Series VPN Configuration Guide." 8 May 2001. URL: http://www.cisco.com/univercd/cc/td/doc/product/core/7100/swcg/index.htm 20 May 2001.

[8] Cisco Systems. "Why Can't I Browse the Internet Using a GRE Tunnel?" Cisco Tech Notes. 2001. URL: http://www.cisco.com/warp/public/105/56.html 20 May 2001.

Additional Sources:

Ferguson, Paul , Huston, Geoff. "What *is* a VPN?" Revision 1. April 1998. URL: http://www.employees.org/~ferguson/vpn.pdf 20 May 2001.

Egorov, Andrew. "Implementing Virtual Private Networks – Observations from the Field." 29 April 2001. URL: http://www.sans.org/infosecFAQ/encryption/implement_VPN.htm 20 May 2001.

Fraser, Moye. "Understanding Virtual Private Networks (VPN)." 3 March 2001. URL: http://www.sans.org/infosecFAQ/encryption/understanding_VPN.htm 20 May 2001.

Cisco Systems. “Sample Configuration: GRE and IPsec with IPX Routing.” Cisco Tech Notes. 2001.
URL: http://www.cisco.com/warp/public/707/33.shtml 20 May 2001.