An Internet-wide Measurement and Security Analysis of IPsec

Andrew S. Prudhomme
A05419855

Brown R. Farinholt
A53044042

Edward L. Sullivan
A53045017

University of California, San Diego

Abstract

Tools that facilitate simple Internet-wide scanning for security vulnerabilities and statistics have existed for years, yet an Internet-scale deployment and security evaluation of the IPsec protocol suite has never been conducted. We develop and conduct a scan of all IPsec end hosts in the IPv4 address space that use the Internet Key Exchange protocol, and use the responses to generate metrics concerning the security configurations chosen by the end hosts. We analyze these metrics, as well as two areas of particular interest: certificates used in authentication, and IKE vendor identification fields, and discuss their security implications.

1 Introduction

In 2013, the ZMap team at the University of Michigan unveiled its eponymous tool, zmap[6]. Capable of scanning the entire IPv4 address space in roughly 45 minutes, and proving to be more adept at large-scale scanning than nmap, zmap quickly became one of the primary tools used in conducting Internet-scale scans. Since its unveiling, zmap has been used to conduct global security surveys of the IPv4 address space, and has resulted in more than one groundbreaking security revelation. Most notable, UMich researchers discovered that the entropy sources of many devices on the Internet were insufficient, resulting in weak (and even overlapping) RSA keys being generated[7].

Despite this proliferation of Internet-scale security scans, most have been focused on TLS/SSL due to its prevalence throughout the web. In fact, not a single survey of notable scale has been conducted on IPsec.

IPsec is a security protocol comparable to TLS/SSL, with several notable differences. IPsec provides security over the network layer, whereas TLS provides it over the transport layer. The implications of this are that IPsec can actually protect its header information, while TLS cannot. Further, operating at a lower layer than TLS, IPsec can provide security to traffic regardless of transport protocol.

While TLS is well suited for dynamically establishing secure communications in a low-trust environment due to its certificate ecosystem and key negotiation per session, IPsec requires significantly more configuration and is better suited for secure connection of pre-authenticated clients and end hosts. For this reason, pre-shared keys (PSKs) are the most common method of IPsec authentication, and its main usage is virtual private networking. IPsec is particularly useful for connecting multiple networks, or remote users to a large network, transparently due to its layer of implementation.

In this paper, we conduct a cursory security overview of IPsec at Internet-scale. We provide technical background on IPsec connections, detail the scanning tools used in this experiment, explain our methodology, analyze and discuss the results we obtain, and discuss future work.

2 Background

2.1 Internet Key Exchange Protocol

In order to aid in the process of establishing a security association, IPsec is capable of using the Internet Key Exchange (IKE) protocol. This protocol can negotiate sessions with two different handshake types, main mode and aggressive mode. A main mode session is negotiated in three phases, shown in Figure 1. The client begins by proposing a set of security transforms. The transforms contain choices such as what authentication, encryption, and hashing schemes to use. The server then returns if one of those proposals is acceptable. Afterwards, a Diffie-Hellman key exchange takes place and the association is finalized. It is then possible to send IPsec traffic to that host.

An aggressive mode handshake combines the transform proposal with the key exchange. This allows for a shortened sequence. It is supported less than main mode and we use it only for certificate collection.

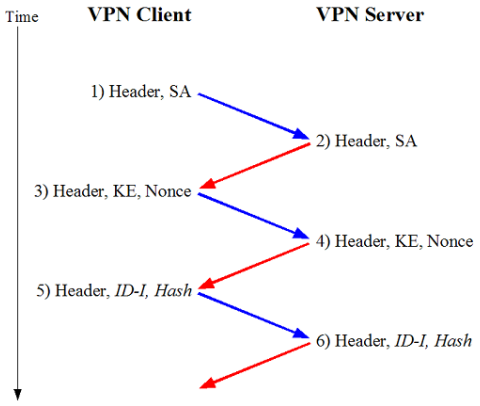

Figure 1: IKE main mode handshake (from ike-scan docs)

```
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
1.236.106.30    Main Mode Handshake returned
        HDR=(CKY-R=3158c22fbc2c0a48)
        SA=(Enc=DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)
        VID=299ee8289f40a8973bc78687e2e7226b532c3b760000000900000500 (Netscreen-01)
        VID=4485152d18b6bbcd0be8a8469579ddcc (draft-ietf-ipsec-nat-t-ike-00)
        VID=4865617274426561745f4e6f74696679386b0100 (Heartbeat Notify)

Ending ike-scan 1.9: 1 hosts scanned in 0.184 seconds (5.44 hosts/sec).  1 returned handshake;
```

Figure 2: Example output from ike-scan

3 Scanning Tools

3.1 Zmap

ZMap is a tool used to quickly and efficiently scan a large collection of host computers. It has modules that allow for TCP, ICMP, and UDP payload scans. Under the best conditions, ZMap is able to probe the entire IPv4 address space in under five minutes. In our study, we use this tool to perform a series of complete Internet scans.

3.2 Ike-scan

Ike-scan is a command line utility that is capable of probing for information from a host's IKE process. It does this by sending out the first packet of an IKE negotiation and parsing the reply from the host. Ike-scan allows you to fully configure various options including the security association transforms proposed to the remote host. The transform includes, among other things, the desired authentication mode, encryption scheme, hashing algorithm, and Diffie Hellman group.

If the host accepts one of the proposed transforms, it sends back a response that is parsed by ike-scan. The response contains the host's transform preference. An example of ike-scan output is shown in Figure 2. This result can optionally contain vendor IDs that reveal additional information about the IKE process or system. We use this tool to collect IKE information to gain statistics on IPsec configuration and usage.

4 Methodology

4.1 Setup

The majority of our scans were carried out by a single VM running Ubuntu. This system had very modest specifications (only a single core). As the study progressed, we added a second VM to cope we the volume of scans we needed to run.

Our primary goal was the collection of information. We had no wish to bother other network administrators or scan those that did not wish to be scanned. To this end, we implemented the standard best practices typically used when conducting Internet wide scans. Our VMs were configured with a reverse DNS entry with the base name "ipsecscanner" to clearly identify our intentions. Additionally, a web server was set up for each system that presented a page explaining who we were and exactly what we were doing in terms of goals and scope. This site also contained an email address to allow the opting out of data collection and future scans. All requests were added to a blacklist used by zmap and any associated database entries were removed to prevent further scanning. We thus attempted to be as courteous as possible while collecting our data.

4.2 Challenges & Limitations

There are some serious challenges to collecting the data we wanted. Given the level IPsec operates on, it is not possible to directly query its existence. There is no direct IPsec session establishment, as there is with TSL. Additionally, even if you know a host supports IPsec, you cannot ask the host what security associations it supports. In order to obtain this information, we must rely on it being revealed though the IKE protocol, which carries its own issues and limitations.

The IKE protocol is not absolutely required to establish an IPsec connection. Though there exists implementations of the protocol deployed by VPN vendors and major operating systems, it is possible to manually configure the security association rather than negotiate it. This would require addition coordination between host and would be much harder to manage. We suspect that these types of connection are a minority of the IPsec usage, however, we cannot know for sure as our scanning will get no response from these hosts.

An additional limitation of an IKE based scan is that we will only be able to obtain information about the anonymous profile. Most IKE implementations allow for different security association options to be configured for different hosts. The host name serves as the profile identifier when negotiating a session. As we are not attempting to impersonate a known host, our IKE negotiation will use the information in the default, anonymous profile. Thus, we will only be able to gain information on

this profile.

Another complication is that there are a large number of different IKE process implementations. This creates issues, as not all of them respond in the same way. The IKE process is not required to send back a response if it does not recognize the host or accept any of the security proposal transforms. The behavior in these situations varies from an error response to no response. Additionally, a positive response only means that at least one of the proposed transforms was acceptable. Some documentation indicates this is the most secure proposal that the host supports, however, it could be implementation specific. We refer to an accepted proposal as the host's preferred configuration among the choices.

This alludes to the major limitation. To get the most complete picture of what security association transforms are supported it would be required to send each possible configuration as an individual IKE request. As there are 120 different combinations of options, this is infeasible. It would take a great deal of time to perform such a collection of scans and the amount of unsolicited traffic sent to each host would be beyond what we consider acceptable for a benign scan. This required us to form transform groups that gave us the most useful information about IPsec deployment, while not being overly intrusive.

4.3 Ethics

There are some potential ethical issues involved with sending unsolicited traffic to external hosts. To resolve this, we established the following guidelines for our scans. First, we did not make any effort to complete an IKE handshake. We only sent the initial packet and observe the host's response. Additionally, we limited the volume of traffic sent to any host. We did not attempt to brute force the security transforms.

4.4 Scanning

We performed our scanning in two phases. The first used zmap to create a list of IP addresses that potentially supported IPsec. The second used this list to collect information on the hosts' IPsec configuration using ike-scan.

4.4.1 Phase 1: zmap

For this study, we performed three complete scans of the IPv4 address space using zmap. Of these scans, two were focused on the pre-shared key authentication mode and one was focused on the RSA signature authentication mode.

Two modes were selected since preliminary testing of an IKE implementation (raccoon) showed that some IKE processes might not respond to proposals with the wrong authentication mode. These particular two modes were chosen since they are considered to be common [8]. This was one of the limitations we used to restrict our scan combinations.

The IKE protocol functions over UDP. In order to use the zmap UDP module, a payload must be provided. We obtained this by using ike-scan against our test IPsec system and recording the traffic using Wireshark [5]. We extracted the payload from one packet for each of our target authentication modes.

Each of the three zmap scans were performed about a day apart. The purpose of doing two scans with the pre-shared key payload was to get a sense of how consistent the response addresses were on the day timescale. We felt this would be important to evaluate the differences in IP addresses returned between the two authentication mode scans.

The scans each took about ten hours to run. The end result was a list of IP addresses that sent back some form of response to our ike-scan payload sent to the IKE UDP port (500). This information was loaded into a MySQL database and used for the second phase of scanning.

4.4.2 Phase 2: ike-scan

The second stage of scanning was intended to collect information on security association configurations used by Internet hosts. To accomplish this, ike-scan was used to probe different sets of security transform combinations. If the host accepted any of the transforms, the output of the ike-scan was parsed with regular expressions to extract the configuration. All options were loaded into a database table, though, the ones of most interest were the authentication mode, encryption scheme, hashing algorithm, and Diffie Hellman group.

Since it was not practical to probe for every transform option combination individually, we created two different groups to scan. The goal was to have large coverage of the common transforms, and yet be able to make security inferences. To this end, we scanned each of the two authentication modes with two different transform groups.

We created lower and higher security groups of transforms. The lower group was composed of eight different transforms. It used DES/3DES for encryption, MD5/SHA1 for hashing and modp768/1024 for the Diffie Hellman groups. These are the most common sets of options deployed and are the default scanned using ike-scan. The higher security group used similar options, but looked for AES encryption of three key lengths (128/192/256). We recorded the maximum key length supported. We also added the option for SHA2-256 to see if anyone was deploying more advanced hashing schemes.

In addition to the accepted security association we also collected any vendor ID information returned with the ike-scan response.

This phase of scanning took substantially longer than the zmap scanning. Doing a multi-threaded scan, the low security scan group took about five days to complete for an authentication mode and the high security group took about seven days (since it had more transform combinations).

4.4.3 Collecting Certificates

One of the authentication modes we chose to scan was RSA signatures. Since this authentication mode requires the use of certificates, we attempted to collect them for security analysis. We found that a host may present its certificate in the first response packet during an aggressive mode IKE handshake. Thus, this method of collection did not violate our "one packet sent, one packet receive" scanning ethical guideline.

We compiled a slightly modified ike-scan binary that would dump any received certificate data to file on disk. We used OpenSSL to convert this binary to a standard PEM format and stored it for analysis. We only attempted to collect a certificate from hosts that had responded positively to a security association proposal with the RSA signature authentication mode.

5 Results & Analysis

We present the results of our scanning, and analyze their security implications.

5.1 Zmap

Each of our three zmap scans returned a list of IP addresses of host that responded to our IKE packet payload. Table 1 shows the number of unique addresses that responded per scan. The table also contains the number of addresses that were in the scan that were not present in the first scan.

	Pre-shared 1	Pre-shared 2	RSA Sigs
Unique IPs	4449377	4362828	4154226
Not in (1)	—	171932	254730

Table 1: IP addresses collected with zmap

The two pre-shared key authentication mode scans are fairly consistent. The second contained less results than the previous scan yet also contained some additional addresses. Overall, 96.1% of the addresses were the same. This showed the baseline variance on a day timescale and provided context for comparison with the third scan.

The third scan was performed using a different IKE authentication mode payload. Though there were more additional addresses returned, 95.6% of the results had already been seen in a pre-shared key scan. This was not substantially more new address than the observed daily change. Thus, even if we had the resources to do scans for other authentication mode variants, we would likely have collected a similar base short list of addresses for use with ike-scan.

Overall, the zmap scans resulted in 4,802,746 unique IP addresses.

5.2 Response Distribution

Our goal was to scan as much of the IPv4 address space as possible. However, zmap's default blacklist excludes 592,708,865 IP addresses reserved for local networks, loopback, private use, broadcast, multicast, etc. Additionally, through about 30 email requests and complaints, we blacklisted another 170,388 addresses, representing 0.03% of our blacklist. As shown in Figure 3, we ultimately scanned approximately 3,702,088,043 addresses representing 86.2% of the IPv4 address space.

As shown in Figure 4, a total of 4,802,746 unique hosts, representing 0.13% of the scanned addresses, responded to a single IKE handshake initiation packet during our zmap scans. Unfortunately, the data are not normalized for the actual population of active responsive machines on the Internet. If we had extra time, we would have performed a zmap scan with ping packets in order to loosely enumerate the population of responsive machines on the Internet. Instead, we use the data reported by the controversial Internet Census Project (ICP) of 2012. The ICP reports that, as of 2012, 420 million addresses in IPv4 space are active according to ICMP echos, of which 165 million have at least one active port open. Thus, of all potentially active addresses on the Internet, we had a response rate of 1.1%, but, if we consider only those hosts that are running at least one service, we had a response rate closer to 2.9%. (It is worth noting that, between 2012 and 2015, IPv4 space likely changed enough to render these statistics approximate at best) Figure 5 shows a geographic heat map of all the hosts who replied.

Of the hosts who replied to our IKE handshake initiation packet, 89.9 % of them replied with a valid IKE notify packet. We interpret this number, 4,318,473, as a lower bound on the number of Internet-accessible hosts that support IKE and IPsec. Our estimate is probably conservative because we did not test every possible, albeit uncommon, authentication method, such as DSS Signature, RSA Encryption, Revised RSA Encryption, El Gamel Encryption, Revised El Gamel Encryption, ECDSA Signature, hybrid mode, or XAUTH. Furthermore, none of our Security Association proposals included rare hashing algorithms like Tiger, SHA2-384, and SHA2-512 or encryption algorithms like IDEA, Blowfish, RC5, CAST, and Camellia. We also excluded most rare Diffie-Hellman groups. A further challenge is that many hosts who support IKE might ignore all anonymous connection attempts or might sit behind restrictive

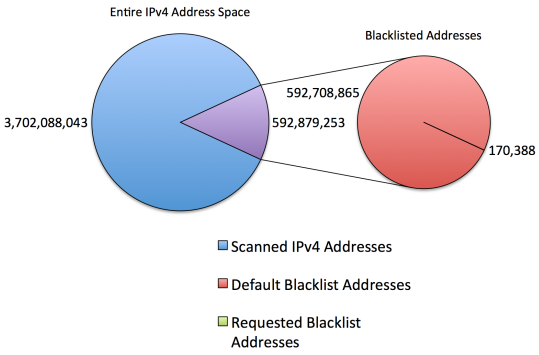

Figure 3: Blacklisted Addresses in IPv4 Scan

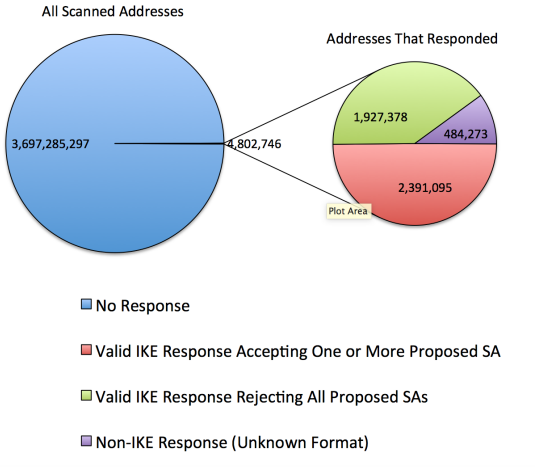

Figure 4: Response Rate of All Scans

Figure 5: Global Distribution of IKE Responses

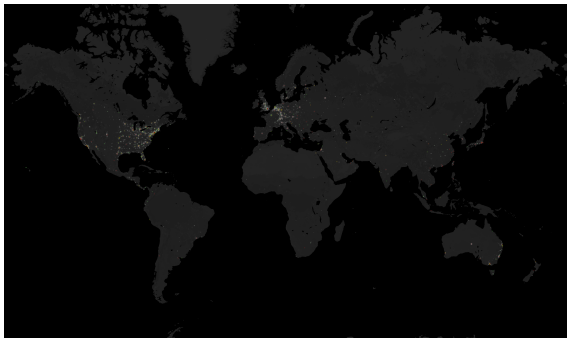

Figure 6: Global Distribution of AES Support

firewalls or NATs. Finally, we do not have a strategy for identifying or estimating how many machines support IPsec but not IKE.

The remaining 10.1% of hosts who responded to our scan sent non-IKE packets of varying formats. We have not attempted to parse or analyze these packets.

Among the hosts who replied to our scan with valid IKE notify packets, about 55.4% accepted at least one of our Security Association proposals. The IPv4 heat map in Figure 7 shows the distribution of these hosts who accepted one or more proposal. Each pixel in the heat map represent a /24 subnet consisting of 256 hosts. The color of the pixel depends on how many hosts in the subnet accepted a Security Association proposal, ranging from dark blue (1-25 hosts) to bright red (230-256 hosts). Black pixels indicate that no host in the /24 subnet accepted one of our Security Association proposals. The addresses are positioned on the heat map according to a fractal pattern called a twelfth order Hilbert Curve such that any consecutive string of IPs will translate to a single compact, contiguous region on the map (XKCD). Unfortunately, as previously mentioned, the data are not normalized for the number of active and responsive hosts on the Internet.

The most apparent observation from the heat map in Figure 7 is its consistently blue and black spattering, which suggests that IKE is widely and sparsely distributed. Zooming in reveals that there are occasional clusters of green or red pixels. Normalizing that data to account for the distribution of active responsive hosts on the Internet might reveal additional hot-spots. We hypothesize that these hot-spots are companies, universities, or large entities that provide IPsec VPN access to all their hosts as part of a standard company-wide policy.

Among the hosts who replied to our scan with valid IKE notify packets, about 44.6% rejected all of our Security Association proposals. There are several possible explanations for these rejections. Some of the hosts might not support anonymous connections. Some hosts might not support any of the specific authentication methods, encryption algorithms, hashing algorithms, or Diffie-Hellman groups included in our particular proposals. Unfortunately, when IKE sends a rejection packet in response to a Security Association proposal, it does not explain why the proposal was rejected or provide a counter proposal. However, given more time and re-

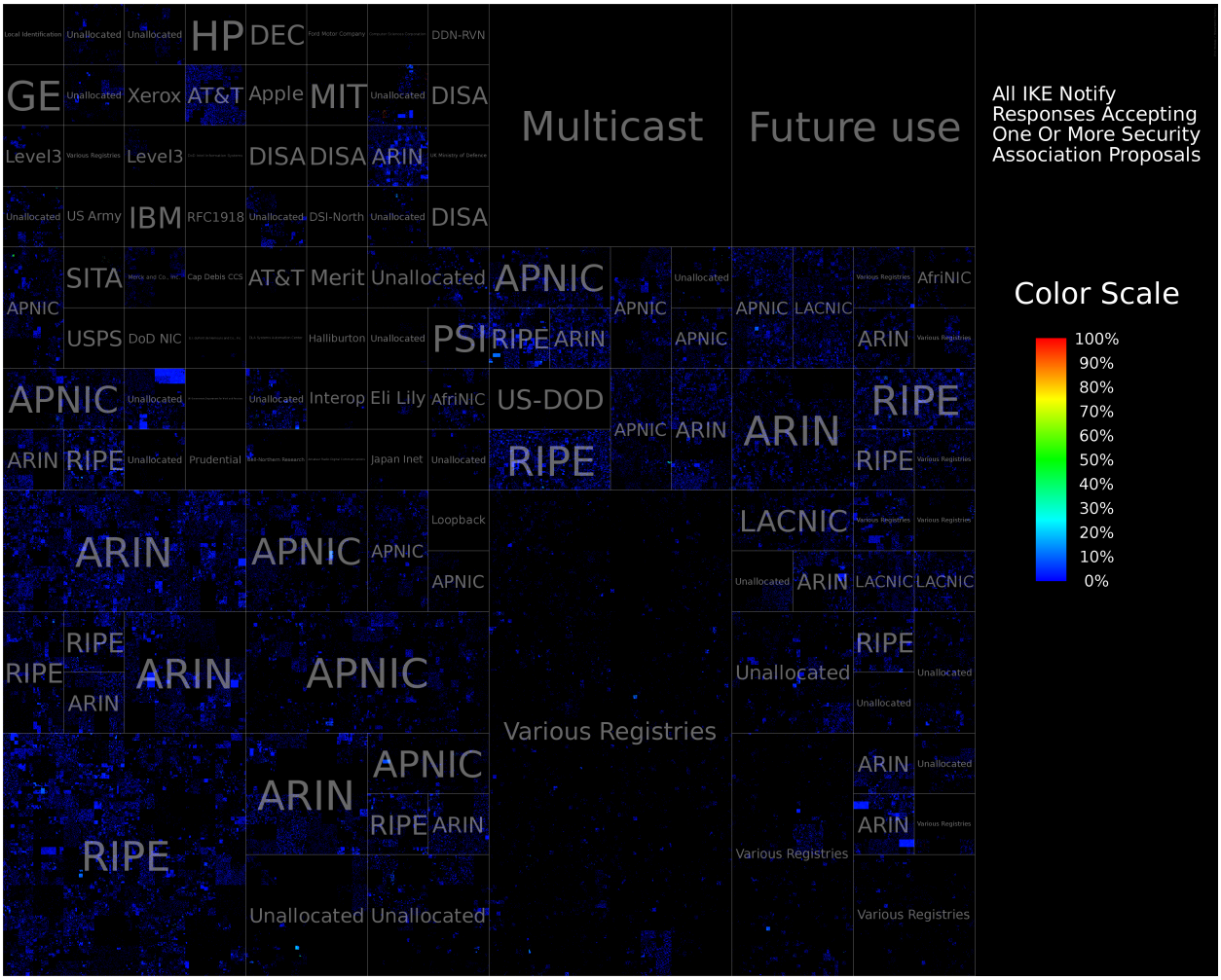

Figure 7: IPv4 Heatmap of All IKE Responses Accepting One or More Proposed SAs

sources, we could have systematically proposed almost all possible combinations of cryptographic suites to determine more accurately the number of hosts willing to accept IKE proposals.

5.3 Configuration Distribution

We performed four major ike-scans: using both Pre-Shared Key and RSA Signature authentication, we sent one Security Association proposal consisting of DES, 3DES, MD5, and SHA1 and another Security Association proposal consisting of AES, MD5, SHA1, and SHA2-256. Rather than provide an exhaustive analysis of all the configuration distributions for all the scans, we will instead highlight the most interesting results.

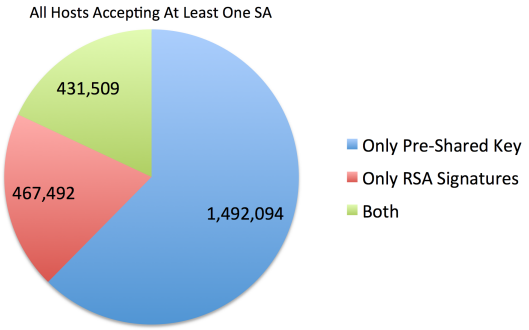

Figure 8: Distribution of Hosts Supporting Various Authentication Methods

5.4 Distribution of Authentication Modes

Although IKE allows for 10 different authentication methods, we only focused on two common methods, namely Pre-Shared Key and RSA Signature. Among the 2.4 million hosts who accepted at least one Security Association proposal during any of our four scans, we found that 62.4% only supported Pre-Shared Key, 19.6% supported only RSA Signature, and 18.0% supported both.

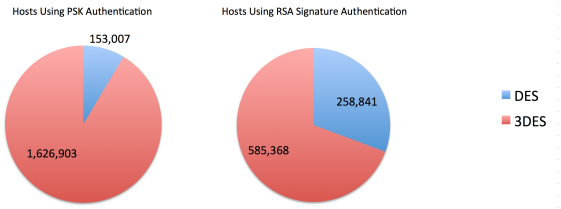

Figure 9: Hosts' Preferences for DES vs. 3DES

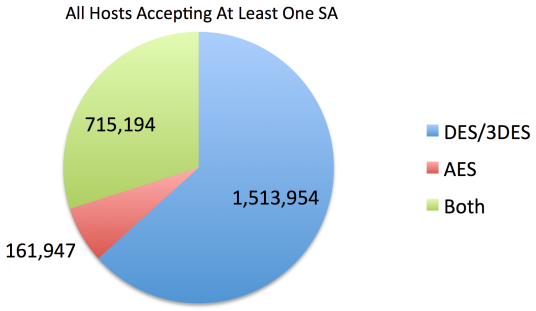

Figure 10: Hosts' Support for DES/3DES vs. AES

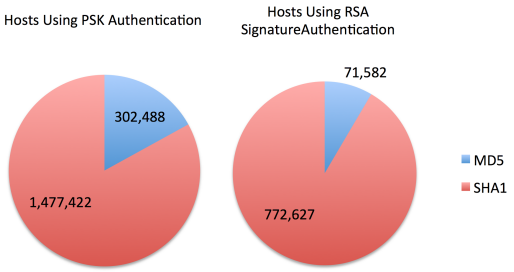

Figure 11: DES/3DES-Supporting Hosts' Preferences for Hashing Algorithm

5.5 Distribution of Encryption Methods

Our first scan included a Security Association proposal consisting of block ciphers DES and 3DES. According the National Institutes of Standards and Technology (NIST), DES is no longer secure and no longer approved as a government standard. However, NIST expects 3DES to remain secure until 2030. Thus we were surprised to discover in Figure 9 that many hosts are configured to prefer DES instead of 3DES. Furthermore, we noticed that hosts using RSA-Signature authentication were more like than hosts using Pre-Shared Keys to prefer DES over 3DES, at 30.7% for RSA Signature and 8.6% for PSK (among the population of hosts that support at least one of the two blocks ciphers). We surmise that this discrepancy exists because using IPsec with with RSA signatures requires a simpler setup that is essentially plug-and-play. In contrast, using pre-shared keys involves a more intentional configuration process and requires more knowledge about cryptography usage. Another theory is that many IPsec hosts prefer DES over 3DES because they were configured before DES became outdated and before 3DES was created (1998).

In our second pair of scans, we attempted to find all the hosts supporting the AES block cipher. The geographic heatmap in Figure 6 shows the 877,141 hosts we discovered who support AES. Although AES is an export-controlled algorithm, we see no obvious evidence in our heatmap that its deployment is actually limited outside the US. However, in Figure 10, we surprisingly see that the DES/3DES block cipher family is much more commonly supported than AES, despite the fact that AES is faster and more secure than 3DES. Our only hypotheses to explain this paradox are (1) that export control is indeed working and preventing the spread of AES, (2) that many hosts prefer using DES/3DES in order to be backward compatible with as many other machines as possible, (3) that many IPsec machines were configured before the creation of AES, (4) that many machines lack hardware support for AES, or (5) that most people poorly configure their machines.

5.6 Distribution of Hashing Methods

During our scans for DES and 3 DES, we tested MD5 and SHA1. During our AES scans, we additionally included SHA2-256 in our Security Association proposals. The results are shown in Figure 11 and Figure 12, respectively.

Although MD5 is usually faster than SHA-1, many attacks have been described for MD5 and several collisions have been found. In contrast, no collision has yet been found for SHA1. Thus, for security purposes, SHA-1 is preferable to MD5. Likewise, SHA2-256 is considered slower and more secure than SHA-1.

Since AES is more secure and faster than DES and 3DES, we would have expected systems smartly configured with AES to also be smartly configured with SHA-1 or SHA2-256. Along the same vein, we expected hosts poorly configured with DES and 3DES to also be poorly configured with MD5. In fact the opposite trend was true: AES-supporting hosts were more likely than DES/3DES-supporting hosts to prefer MD5 as their hashing algorithm. We are not sure how to attribute this strange discrepancy. We know that historic availability is not the cause of this weird behavior since both MD5 and SHA-1 were available by the time AES was created (although MD5 collisions had not been yet found at that time). Perhaps if we were to classify the hosts based on what hardware or software they were using to implement IPsec (by looking for sets of hosts with the same tuple of configuration options and VIDs), it would reveal that widespread reliance upon default product configuration options is the source of this puzzling phenomenon.

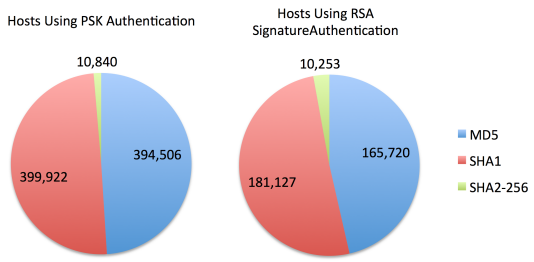

Figure 12: AES-Supporting Hosts' Preferences for Hashing Algorithm

5.7 Certificates

Our scan of 901,000 end hosts supporting aggressive-mode and using RSA signature authentication returned 4,783 certificates, a return rate of about 0.5%. A rate of return this low was expected, as RSA signature authentication is an uncommon choice for an IPsec VPN tunnel, and aggressive-mode entails elevated potential risk with only minor return in terms of negotiation speedup. However, given such a small sample size, all conclusions drawn based on these certificates must be regarded as potentially non-representative.

Two aspects of x.509 certificates are of particular concern with regards to IPsec security: RSA public key length, and certificate expiration date. (There are other areas of concern, particularly the certificate authority ecosystem and the entropy of the certificates' key material, but analyzing these requires data beyond the scope of our scanning, and is reserved for future work.)

5.7.1 RSA Public Key Length

RSA public key length is directly proportional to the security of the connection established using said key in RSA encryption. The longer the key, the more difficult it is to factor it into the two prime numbers that are used to generate the secret key material. If a public key is factored and these two primes are recovered, then the private key is trivially computed, and the public/private key pair is no longer able to provide any security to communications whatsoever.

Over time, as available computational power increases, factoring RSA keys of increasing lengths becomes more and more feasible. As such, NIST has established a time line dictating when keys of various lengths will cease to provide adequate protection to their users, meaning they are assumed factorable. [2] Per this time line, 1024-bit keys were deemed insecure in 2010, as many cryptographic authorities believe that factoring a 1024-bit key has been possible since then given the proper computational power and strategy. Similarly, 2048-bit keys are anticipated to lose their potency in 2030. Given these recommendations, we examine the distribution of key lengths found in the recovered certificates, shown in Table 2.

Key Length (bits)	Distribution	Percentage
512	1	<0.1
1024	3859	80.7
2048	910	19.0
3072	1	<0.1
4096	12	<0.1

Table 2: Distribution of RSA Public Key Lengths

Comparing this table to the NIST recommendations, we find that over 80% of the certificates received have insufficient key lengths, and are therefore theoretically incapable of establishing secure communication as they can be factored. One certificate in particular provides a 512-bit key, which can, as of 2015, be directly factored within hours at a cost of $9000. [1] However, short key length alone is not an entirely damning factor; we must also examine certificate expiration date.

5.7.2 Certificate Expiration Date

Often, the strategy used to obsolete certificates with insufficient key lengths is setting a short certificate expiration date. The expiration date, or "not-after date," is a field in the x.509 certificate in which the certificate's creator can specify the date after which the certificate should no longer be accepted as valid. By setting a certificate to expire before its key length goes "legacy" by NIST standards is a way to prevent insecure keys from being used in communications after they have been broken. Other methods also exist to prevent the use of compromised certificates, namely the certificate revocation list (CRL), but the CRL is reactive by nature. In fact, NIST guidelines also suggest expiration dates as short as three years, for the same reason that passwords should be changed frequently. We examine the expiration dates of all received certificates, and compare these to the certificates' corresponding RSA key lengths, in Figure 13.

The first thing to note about Figure 13 is the number of expired certificates in the ecosystem. Of the 4,783 certificates examined, 103, or 2.15%, had already expired, yet were still being served in the IKE handshake. Once the "not-after" time of a certificate passes, it should no longer be accepted as a valid form of identification. An expired certificate could be compromised, or may be laden with a key that is not longer sufficient, but the main reason to not accept an expired certificate is that, upon expiration, it is removed from the CRL, so its revocation status cannot be checked. Expired certificates should never be accepted for this reason, as their compromise would be completely invisible.

In Figure 13, we also see the enormous portion of certificates using 1024-bit keys. What is notable about these

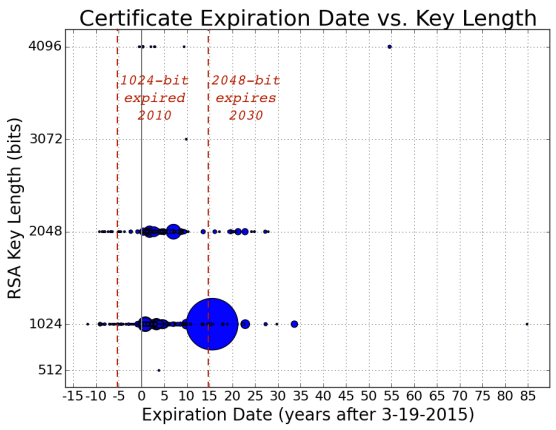

Figure 13: x.509 Certificate Expiration Date vs. RSA Public Key Length

certificates is that the vast majority of them have expiration dates well beyond 2010, the official end of 1024-bit key security. In fact, the majority of these certificates actually expire beyond 2030, the date when 2048-bit key security should expire! One certificate in particular does not expire until the year 2100. An obsolete key length, combined with an expiration date far past the time of the key's obsolescence, begets a certificate with virtually no security against a powerful or determined adversary.

While on the subject of certificates with 1024-bit keys and protracted expiration dates, we note the enormous bubble in the graph at key length 1024-bit, expiration date 10-10-2030. There are 752 certificates represented at this point, received during the scan from 208 different organizations. Upon further investigation into these certificates, it appears each of these certificates is served by a Barracuda IPsec VPN Firewall. The x.509 fingerprint of this VPN configuration is "O=Barracuda Networks, Inc, CN=Barracuda Firewall." Barracuda IPsec VPNs do, in fact, support RSA signature authentication for client-to-site connection by default, and provide tools for creating certificates automatically.[3] Per this data, especially the number of different organizations with the same configuration, it can be inferred that the Barracuda IPsec VPN certificate generator defaults to a key length of 1024-bits, and an expiration date of 10-10-2030. This is an insecure configuration, and should not be the default in any situation. (The exception to this being if Barracuda IPsec VPNs run on hardware that does not support 2048-bit keys or higher; however, this would imply total insecurity of said VPNs against capable threats.)

A final security concern in Figure 13 is that of the protracted expiration dates of the 2048-bit keys. While 20% of the certificates observed have keys that are, at present, thought to be sufficiently secure against factoring, NIST has determined that 2030 will mark the advent of factoring of 2048-bit keys. Roughly one-third of the certificates with 2048-bit keys have expiration dates beyond 2030, leaving them open to attacks in the future; however, these certificates can be revoked whilst not expired, so this is more of a concern than an actual security vulnerability for the time being. But, given the trend with 1024-bit keyed certificates, 2048-bit key certificate expiration dates should be monitored closely as 2030 approaches.

5.8 IKE VID Field

The IKE handshake contains multiple fields for conveying identifying information between the authenticating parties. One of the fields is the vendor identification (VID) field. The VID field is intended, per IETF standards, for conveying several types of information important to negotiation and configuration of an IPsec connection: XAUTH interoperability information[10], NAT-traversal information, and checksum requirement information[9].

Of the 2,391,095 end hosts that returned valid IKE handshake information, 1,357,688 responded with known VIDs, or 56.8%. In these responses, we found a total of 3,619,334 recognized VIDs. Table 3 contains a breakdown of these responses.

Vendor ID	Distribution
Dead Peer Detection	1159352
draft-ietf-ipsec-nat-t-ike	841599
IKE Fragmentation	730812
XAUTH	252743
RFC 3947 NAT-T	192623
Windows	180324
Cisco Unity	102527
strongSwan	59986
Netscreen	21264
KAME/racoon	1790
FortiGate	453
StoneGate	366
Firewall-1	131
OpenPGP	14

Table 3: Distribution of Vendor IDs Returned

There are several interesting observations to be made with regards to the VIDs received. First, there is far more information in this chart than simply XAUTH and NAT-T configuration information. This is because the VID field has largely been re-purposed to include any system configuration information the proprietor of the IPsec end host wishes to send to potential clients. We observe that some of this information is not only non-adherent to the original purposes of the VID, but is in fact irrelevant to the IKE handshake.

IKE Fragmentation VIDs, for example, mean that the

packet is part of a fragmented handshake response that was too large for a single handshake packet. This signifies fairly drastic modification of the IKE handshake process by end hosts.

Further, the Windows VIDs, expanded upon in Table 4, reveal the operating system being used by the IPsec end host. Not only is this information not relevant to the IKE handshake process whatsoever (the IKE implementation VIDs handle per-system configuration details), it actually allows the system to be targeted by attackers with operating system-specific exploits. Leaking system configuration information unnecessarily is always dangerous, especially when the operating system being run is no longer being supported by Microsoft and has significant, known vulnerabilities.

Windows Version	Distribution	Percentage
MS NT5	105017	58.2
Win 2003, XP-SP2	74608	41.4
Win 2000	698	0.4
Win Vista	1	$<$0.1

Table 4: Distribution of Windows OS Versions

Though the VID is intended to convey information for XAUTH clients, we did not attempt to connect to any end host using XAUTH authentication, and yet over 250,000 end hosts replied with a VID that implies XAUTH support. There are several known attacks against implementations of XAUTH IPsec[4], and revealing support for an unsafe authentication mode to untrusted clients seems risky.

As a final note on VID information leakage and potential vulnerabilities, in 2010 an attack on Cisco IPsec end hosts was discovered, in which the group ID, a secret given only to clients for authentication, could be brute forced. When a client initiated a handshake with the end host using the correct group ID, the end host would return a VID corresponding to its Dead Peer Detection (DPD) version; if the client had the wrong group ID, no DPD version VID would be served. This was exploited to brute force the required group ID. Attacks such as these are why VID information leakage is not harmless.

6 Discussion & Future Work

6.1 SA Configuration

In our study, we collected data on a limited set of security transform groups. This was due to limited time and resources, combined with the practical/ethical considerations of performing Internet scans. A more complete set of data could be obtained by scanning for each transform configuration individually. This would require on the order of hundreds of scans. Additionally, we have shown that there is some change in the IPsec host IP addresses at a day timescale. A complete picture would require that these dynamic hosts be correlated across many scans at different times. We leave data collection of this detail as future work.

6.2 Certificates

We have found that, for the small sample of certificates we were able to recover, their security provisions are lacking at best. With 80% of certificates using insufficient keys, and protracted expiration dates for an even larger portion, the measured certificate ecosystem is providing inherently weak security to those IPsec end hosts supporting RSA signature authentication.

For future work, we plan to compare the certificates to the CRL, to follow their certificate authority chains, and to develop a method for extracting certificates from IPsec end hosts that do not support aggressive mode, but still support RSA signature authentication.

6.3 Vendor IDs

As we have pointed out already, the Vendor ID field has been re-purposed into an assorted configuration information field. While it has legitimate purposes, it is often used to convey sensitive system configuration information, a practice which must be eliminated to improve the security of IPsec end hosts. The IETF has already called for the elimination of the VID field[10], replacing it with specific fields for the information VIDs are intended to convey. We support this.

For future work, we would like to explore the VID choices of specific implementations of IKE, and determine whether more attacks akin the the Cisco brute force group ID attack are possible.

7 Conclusion

We have conducted a cursory, Internet-scale security scan and analysis of IPsec. We have discovered several issues with IPsec implementation, including certificate malformation and vendor ID information leakage, and have found that many IPsec end hosts support configurations of questionable security. A more in-depth scanning and analysis is recommended, given these findings in such a limited scan. Overall, we have determined that IPsec may not be providing the security benefits that it could due to poor implementation choices, and well as flaws in the handshake process.

References

[1] ALBRECHT, M., PAPINI, D., PATERSON, K., AND VILLANUEVA-POLANCO, R. Factoring 512-bit rsa moduli for fun (and a profit of $9,000).

[2] BARKER, E., BARKER, W., BURR, W., POLK, W., AND SMID, M. Recommendation for key management-part 1: General (revision 3). In *NIST special publication* (2012), Citeseer.

[3] BARRACUDA. How to Create Certificates for a Client-to-Site VPN. `https://techlib.barracuda.com/display/CP/How%2Bto%2BCreate%2BCertificates%2Bfor%2Ba%2BClient-to-Site%2BVPN`, 2013.

[4] CISCO. cisco-sa-20050406-xauth. `http://www.cisco.com/c/dam/en/us/support/docs/csa/cisco-sa-20050406-xauth.html`, 2005.

[5] COMBS, G., ET AL. Wireshark. *Web page: http://www. wireshark. org* (2007), 12–02.

[6] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security* (2013), Citeseer, pp. 605–620.

[7] HENINGER, N., DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Mining your ps and qs: Detection of widespread weak keys in network devices. In *USENIX Security Symposium* (2012), pp. 205–220.

[8] IKE-SCAN. Ike-scan User Guide. `http://www.nta-monitor.com/wiki/index.php/Ike-scan_User_Guide`, 2009.

[9] KIVINEN, T., SWANDER, B., HUTTUNEN, A., AND VOLPE, V. Negotiation of nat-traversal in the ike. Tech. rep., RFC 3947, January, 2005.

[10] PEREIRA, R., AND BEAULIEU, S. Extended Authentication Within ISAKMP/Oakley. `https://tools.ietf.org/id/draft-ietf-ipsec-isakmp-xauth-06.txt`, 1999.