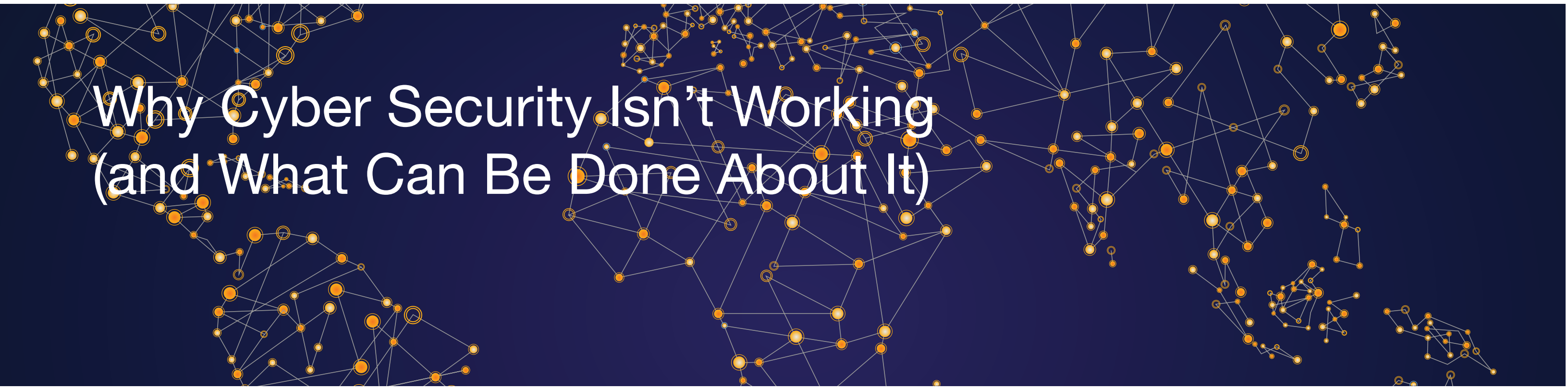
CENTRIPETAL
NETWORKS

Why Cyber Security Isn't Working (and What Can Be Done About It)



A Technical White Paper
Centripetal Networks, Inc.

www.centripetalnetworks.com

05/04/15

Why Cyber Security Isn't Working?

A Technical White Paper
Centripetal Networks, Inc.

Introduction

Since the 1990's, the Internet has rapidly evolved to become part of the core infrastructure for commerce, government, military defense, utilities, entertainment and social life. It interconnects anybody to any computer located anywhere – but unfortunately, anybody includes adversaries who want to harm, steal, or misuse assets belonging to others. Motives vary across the spectrum from misguided curiosity to undermining or even attacking enemy governments.

As fast as the Internet has evolved, the cyber threat has evolved even faster. By 2009, the cyber threat grew large enough to cause the US Department of Defense to declare cyber as another domain of defense – joining land, sea, air and space – which led to the creation of US Cyber Command (USCYBERCOM). Now, the Department of Homeland Security (DHS) is partnered with USCYBERCOM in defending US civilian Internet infrastructure, which makes sense because approximately 90 percent of military networks are dependent on commercial Internet infrastructure. Across US defense, government, and civilian/commercial sectors, US cyber security spending is enormous, measuring in the many tens of billions of dollars each year.

And yet, despite all of this cyber security spending, it appears that the adversaries are winning, and winning big. For example, Gen. Keith Alexander, director of the NSA and commander of USCYBERCOM, estimates that cyber-crime costs exceeded $1 trillion in 2008, and that costs have grown rapidly since then1. Furthermore, the cyber domain is highly asymmetric: a few individuals with cheap, easily obtained tools can inflict massive damage and financial losses, and cause huge breaches in national security, even on strongly defended networks using the most sophisticated, expensive cyber defenses. The most secure, mission critical networks in the world have been breached, even so-called air-gapped networks that do not have any direct connections to the Internet.[1]

Clearly, conventional cyber security methods are not working. Why? And is there anything we can do about it?

To mount effective defenses against the cyber threat, we must first understand why conventional cyber security methods are not working. This white paper provides some insights into the answer, and suggests some potential new paths towards efficient cyber security that can reverse the asymmetry. These insights are presented in a pragmatic, accessible way intended for both non-experts and expert who want to understand why cyber attacks continue to be successful even though many billions of dollars are spent to defend against them.

Internet Cyber Attacks & Conventional Defenses

At the highest level of the cyber threat hierarchy are two main categories of attacks: Denial-of-Service (DoS) attacks, and malware attacks. In a DoS attack, a network resource – a web server, a router, an Internet access link, etc. – is flooded with so much resource-consuming attack traffic that legitimate traffic is effectively denied service by the resource. In a malware attack, a resource is tricked into executing applications that cause damage to, steal from, or otherwise misuse the resource.

Denial-of-Service (DoS) Attacks and Defenses

Analyses of representative DoS attacks and malware attacks will show the difficulties of defending against them using conventional methods. It will also reveal some potential new approaches to defense.

DoS Attack Example #1: Network Bandwidth-Consuming Attack

An example of a network bandwidth-consuming DoS attack is a so-called UDP flooding attack. This type of DoS attack is often carried out by a botnet, which is a large (102 – 106) collection of hosts (distributed across the Internet) that have been zombied: infected with

[1] Gen. Keith Alexander, Military Information Technology, Vol. 14, No. 10, November 2010.

malware that is remotely controlled by an adversary. It is simple for any adversary to access and use a botnet because there are many botnet service providers who have spent years building up large botnets and are eager to rent them out.

The life-cycle of a botnet-driven UDP flooding attack is described as follows: Upon command from the adversary, each zombied host begins sending User Datagram Protocol (UDP) packets with random port values to a target, e.g., an online banking web server. Each zombied host may send a relatively low volume of UDP packets in order to avoid detection by local security mechanisms. Because the size of the botnet can be quite large, the aggregate of many low-volume attacks can be huge.

Adversaries prefer UDP to TCP for many DoS attacks primarily because UDP is connectionless, i.e., a transport level connection between the source (attack) host and the destination (target) host does not need to be established. From the adversary’s perspective, using the connectionless UDP transport protocol vs. connection-oriented TCP has several advantages, with the most important being that setting up a TCP connection requires that the IP address of the connection initiator (the zombied host) be included in the attack packet’s header information – which means that the identity of the attacking host is known.

Conversely, a UDP packet does not require that the zombied host’s IP address be included in the packet header in order for the packet to be transported by the Internet to the target. Thus, adversaries spoof the source IP address in the UDP attack packet, either by forging an IP address of another host, or by using a bogon address. A bogon address is a bogus IP address that is selected from unassigned regions of IPv4 or IPv6 address space. When a UDP packet is spoofed, not only is the actual source concealed, but also it is impossible to trace back to the source using the Internet’s routing tables – and thus, not even the target’s Internet Service Provider (ISP) is able to shut down the attack because they do not know where to block the attacking traffic. In summary, adversaries love bogon addresses and use them whenever possible.

Returning to the DoS attack life-cycle: Assume that the botnet is of sufficient size to sustain a high-volume, high intensity attack. Multiple network resources may be forced to deny service to legitimate users:

1. As the attack packets traverse the Internet and approach the target web server, the resulting implosion of packets consumes all of the capacity of the network links, routers and switches that are close to the target web server. Router/switch packet buffers overflow, causing most legitimate packets to be dropped. Drops of legitimate packets that are part of TCP sessions – i.e., all legitimate web traffic destined for the web server – cause TCP to retransmit the dropped packets, further exacerbating the congestion.

2. The target server’s processing resources are overwhelmed by having to respond to all of the attacking UDP packets. For each UDP packet, the target server must check if a local application server is listening to the port. If there is no application server listening to the port – which is highly likely when the UDP packets have randomly generated port values – then the target server replies with an ICMP Destination Unreachable packet. This not only uses up processing resources but also increases the volume of outbound traffic on the network links, switches, routers and appliances that are close to the target web server.

Regarding this second item, if a network firewall is placed in front of the web server that only allows TCP packets destined for the web server’s port 80 (the HTTP service) or port 443 (the HTTPS service), then the DoS attack packets will be blocked, and thus the web server will be protected. But, this firewall defense is no problem for the adversary who simply sends UDP packets disguised as TCP data packets (by assigning port 80 or port 443 to the TCP header’s destination port field, and TCP to the IP header’s protocol field). Firewalls are not able to figure out that these packets are fakes, and will allow them to proceed to the web server. Then the web server is (heavily) burdened with figuring out that the packets are fakes – and the denial-of-service attack remains successful.

Conventional Bandwidth-Consuming DoS Attack Defenses

The Short Story: There have been no effective defenses for most bandwidth-consuming DoS attacks. This is why they are so popular. Of course, there are several defenses

promoted by the cyber security industry. Below, we review them and briefly analyze why they don't work. But first, let's discuss a somewhat obvious potential defense for many DoS attacks – filtering packets for spoofed addresses – and why this defense is not effective. Recall from above that spoofed addresses come in two forms: bogons and forgeries.

Conventional Bandwidth-Consuming DoS Attack Defenses

The Short Story: There have been no effective defenses for most bandwidth-consuming DoS attacks. This is why they are so popular.[2] Of course, there are several defenses promoted by the cyber security industry. Below, we review them and briefly analyze why they don't work. But first, let's discuss a somewhat obvious potential defense for many DoS attacks – filtering packets for spoofed addresses – and why this defense is not effective. Recall from above that spoofed addresses come in two forms: bogons and forgeries.

Bogons and Bogon Filtering

Bogon addresses should be straightforward to determine: Track/monitor which regions of IPv4 and IPv6 address space have been allocated (by IANA or Regional Internet Registries), and compute the difference from the entire IPv4 or IPv6 address space. In fact, there are organizations that do exactly this, notably Team Cymru Research NFP[3], and then make the lists of bogon address ranges freely and openly available.

As of October 2012, there are nearly 5000 IPv4 bogon address ranges, and more than 61,000 IPv6 bogon address ranges[4]. There is the problem: Filtering packets for bogon addresses require filter rule sets with at least as many rules as bogon address ranges. The most powerful (and expensive) network firewalls typically have practical upper limits of 10,000 rules that can be applied. Similarly, the most powerful (expensive) routers have similar upper limits for their access control lists (ACLs), although in theory up to 80,000 rules can be loaded into high-end router ACLs' TCAM memory (which is fast but very expensive and inflexible). So while it is possible for highend router ACLs to filter all current bogon ranges, (a) soon the number of bogon ranges will exceed the upper limit, and (b) there are other important filtering applications that use router ACLs, so not all of the ACL capacity can be used up by bogon filters. Even if the rules are restricted to filtering on only IPv4 bogons, a 5000-rule set in a firewall or a router ACL will significantly impact network performance (as measured by latency and packet loss). Also, IPv6 bogon address ranges change continually – each time a new region of IPv6 address space is assigned – so keeping up with the changes means frequent updates to the rule sets. Frequent updates not only incur significant operational costs, but particularly in the case of router ACLs, they also cause temporary loss-of-service because router interfaces typically need to be rebooted to load the new rules. For these reasons of scale and adaptability, as well as others, comprehensive bogon filtering is rarely, if ever, performed in practice.

Forgeries and Ingress Filtering

A forgery occurs when a host creates a packet with a source IP address different than the host's actual IP address, i.e., the packet has been spoofed. Hosts forge packets to conceal the hosts' identities. When the forged address is not a bogon, there is no native, straightforward method for determining if a packet's source IP address value has been forged. There is an indirect method called ingress filtering which ISPs may use to detect when spoofed packets ingress their networks[5]. It is based on a router's reachability, the fact that there are a limited number of legitimate source addresses that can be reached by any given router. Briefly, an ISP that deploys ingress filtering determines, for each router at the edge of its network, which IP addresses can be reached by that router. Corresponding filter rules are created and (typically) loaded into the router's ACLs to check each

[2] At any time, there are at least several hundred, and probably thousands or even hundreds of thousands, of active DoS attacks. See, for example, atlas.arbor.net/summary/dos for a report on some current DoS attacks and their characteristics. Verisign estimates that on average there are 130,000 DoS attacks per day.

[3] www.team-cymru.org

[4] Note that because most of IPv4 address space has been allocated, the IPv4 bogon ranges will not change much. In contrast, IPv6 addresses are being allocated on a near continual basis, so the number of IPv6 bogon address ranges is growing rapidly.

[5] RFC 2827 "Network Ingress Filtering: Defeating Denial-of-Service Attacks which employ IP Source Address Spoofing", also known as IETF Best Current Practices
(BCP) #38.

packet ingress the network through the router. If a packet has an unreachable source IP address, then it is blocked[6].

The problem with ingress filtering is it must be widely deployed by ISPs across the Internet to be effective. Because of potential network performance problems (due to filtering), router ACL management costs, periodic loss-of-service due to list updating, lack of incentives, and lack of enforcement, many ISPs do not deploy ingress filtering – even though ingress filtering has been designated by the IETF as a Best Current Practice for ISPs. Accordingly, a study conducted in 2009 estimated that more than 50 percent of Internet-attached hosts can successfully spoof packets[7]. Thus, using spoofed packets in DoS attacks remains a popular and effective technique. And, because the ingress filtering architecture is ISP centric, enterprises receive no protection from DoS attacks using spoofed packets if they deploy ingress filtering at the edge of their networks. It is fair to conclude that this cyber security method is not working.

Firewalls

Most enterprises place firewalls at or near their networks' Internet access points (the locations where they physically connect their networks to their ISPs' networks)[8]. Firewalls are typically configured with packet filtering rules that only allow certain unsolicited traffic into the network, such as traffic destined for the enterprises' public web servers. Accordingly, enterprises' public web servers are frequently the intended target of many DoS attacks.

As the example above shows, it is simple to trick a firewall into allowing DoS attack packets destined for the resources that the firewall is explicitly attempting to protect. Bogon filtering cannot be used because the number of filter rules is much too large for conventional firewalls to apply, and the operational costs of keeping up with the continual changes are too high. Ingress filtering for spoofed packets is not applicable at an enterprise's access points because, except for bogons, every IP address is reachable, and thus forgeries cannot be detected. Even if ingress filtering were somehow applicable at enterprise access points, the size of the ingress filters would be much too large – hundreds of thousands of rules – for conventional firewalls to apply.

Router Access Control Lists (ACLs)

Routers used by ISPs are equipped with packet filters, called Access Control Lists (ACLs), on the router's network interfaces. For ISPs, router performance, and by extension, network performance (as measured by latency and packet loss), is critical to the ISP's business success. Thus, router ACLs are typically implemented in programmable hardware called TCAM[9] so that packet-filtering performance is maximized. Even with TCAM, however, ISPs need to balance the number of ACL filter rules (the amount of cyber security they are providing) with the reduction in network performance. Also, because changing rules in TCAM typically requires a router interface reboot (and therefore temporary loss-of-service, which is anathema to an ISP's business), ISPs are reluctant to use rule sets that require frequent changes in order to be effective. As a result, many ISPs use their routers' ACLs for applications other than cyber security (e.g., rate-limiting), or they use them to provide limited protections via small, mostly static rule sets.

As described above, comprehensive packet filtering for DoS attack defense requires both bogon address filtering and ingress filtering, resulting in rule sets with 100,000 rules or more. This is far too many rules for a conventional router ACLs to apply without significantly degrading network performance. Also, bogon address ranges and ingress filtering rules need to be updated frequently, making the management and operational costs of DoS attack filtering very high. As a result, ISPs do not offer comprehensive DoS attack protection, even though a highly effective method for defending against DoS attacks is filtering the DoS attack packets when they ingress ISPs' networks.

[6] Ingress filtering is not foolproof. If a spoofed packet's forged source IP address is in the same reachability range as the actual source address, then the spoof will go undetected.

[7] "Understanding the Efficacy of Deployed Internet Source Address Filtering", by R. Beverly et al., CAIDA, 2009, http://www.caida.org/publications/papers/2009/imc_spoofer/imc_spoofer.pdf

[8] A notable exception is universities, especially research universities. Research universities want to foster collaboration between their researchers and researchers from many other institutions. Trying to manage network access for a global research community via network firewalls is an impossible task. It is much simpler for universities to eliminate network firewalls and focus instead on host-centric security.

[9] .Ternary Content-Addressable Memory

Intrusion Detection/ Prevention Systems (IDS/IPS)

IDS/IPS[10] systems are designed to detect malware – they are not designed to detect DoS attack packets. While it may be possible to configure an IPS to detect DoS attack packets, the processing requirements during an attack would quickly overwhelm the IPS processing capacity, effectively denying service to legitimate traffic. An IPS is often equipped with a firewall, but these firewalls are no better than standalone firewalls at filtering DoS attack packets.

Over-Provisioning

Some large enterprises with large e-commerce businesses and/or large public clouds and data centers attempt to protect their assets from flooding DoS attacks by massively over-provisioning their Internet access links, i.e., they buy far more access bandwidth from their ISPs than is necessary to service legitimate traffic. Not only is this expensive, and therefore only available to the largest enterprises, but also it is a losing strategy. Flooding DoS attacks with intensities of over 100G have been reported since 2010[11]. To put that in perspective, 100G is ten times the size of most Internet backbone pipes. While somewhat effective in years past for large enterprises that could afford it, over-provisioning is no longer an effective strategy for defending against DoS attacks.

Scrubbing Services

Enterprises may contract with scrubbing service providers to mitigate attacks. A scrubbing service works as follows: When an enterprise is experiencing a flooding DoS attack, all of the enterprise traffic is re-routed to one or more scrubbing centers. These centers employ multiple, often-proprietary techniques to remove DoS attack packets, and then route the remaining packets back to the enterprise. It appears that scrubbing service providers are successfully growing their markets, attesting to the efficacy of these services in mitigating many DoS attacks. But, they are not 100 percent effective, they can't handle the largest attacks, they're very expensive, and they effectively disable any low-latency, real-time applications such as enterprise IP telephony and videoconferencing services.

DoS Attack Example #2: HTTP GET Attacks on Web Servers

The bandwidth-consuming DoS attack of Example #1 was the most common type of DoS attack until 2010. The HTTP GET DoS attack, which directly attacks public-facing web servers, suddenly became quite popular with cyber attackers. Today it accounts for 80-90 percent share of all Internet-borne DoS attacks.

What is an HTTP GET attack, and why is it so popular?

When a web browser is pointed at a web site, say www.somewebsite.com, the browser sends an application level HTTP packet with a "GET www.somewebsite.com" method in it to the web site. This is a request that the web site server sends the www.somewebsite.com/index.html page – the home page – back to the web browser client. The web site handles the GET request by sending back the requested page. In a typical HTTP GET attack, a botnet command & control node commands its many bots to send HTTP GET requests to some target web server. The web server not only bogs down from trying to handle all the requests, but also legitimate HTTP GET requests are starved, thereby causing the denial-of-service.

Some of the reasons that HTTP GET attacks are so popular nowadays include:

1. To be successful, HTTP GET attacks typically require much less bandwidth than a bandwidth-consuming DoS attack, which means that a relatively small botnet can generate enough traffic to successfully attack. Approximately 1G of HTTP GET request/response traffic is enough to bog down a high-capacity web server. Also, 1G usually will not trigger any alarms at the ISP providing Internet access for the targeted web server. The (expensive) bandwidth over-provisioning tactic used by some enterprises to defend against bandwidth-consuming DoS attacks is ineffective against an HTTP GET attack;

[10] The difference between an IDS and an IPS is that an IDS is intended to work offline on stored data to detect malware; whereas, an IPS is intended to work inline and in "real-time" on live traffic.

[11] Network Infrastructure Security Report, Arbor Networks, 01-Feb-2011, www.arbornetworks.com/report

2. HTTP GET attacks are hard to detect – and therefore hard to mitigate/shut down – because of the difficulty of discriminating between malicious HTTP GET requests generated by bots and legitimate HTTP GET requests generated by human-operated web browsers. In fact, for a single HTTP GET request, there is simply no way to tell if it is bot-generated or browser-generated. Instead, discrimination requires analysis of request timing patterns and URL access patterns over many HTTP GET requests. In turn, such analysis requires maintaining historical state information and inspecting packet contents (known as deep packet inspection, or DPI). Because network firewalls and router ACLs neither maintain state nor inspect packet contents, they are completely ineffective at detecting and defending against HTTP GET attacks;

3. Most enterprises' e-commerce services and information services use web-based (web browser) access, including the new generation of cloud services. Hence, an HTTP GET attack botnet has access to every Internet web site and can cause a tremendous amount of economic damage with only a small amount of effort and resources. In a cyber world of asymmetric threats, the HTTP GET attack stands out as one of the most asymmetric;

4. Botnet technology advances make it almost impossible to determine the actual director of an attack. Even though HTTP GET attack bots cannot spoof their host's source IP addresses[12], the machines hosting the bots are usually zombies, under the control of some unknown botnet director, and unaware that they are participating in attacks. Furthermore, even if there are effective methods for detecting HTTP GET attack bots, it is impractical to block them using packet filters because there are so many of them. Conventional filtering technologies – network firewalls and router ACLs – have neither the scalability nor the agility to keep up with the onslaught of bot IP addresses that need to be blocked.

Conventional Defenses against HTTP GET Attacks

As with bandwidth-consuming DoS attacks, today there are no effective, efficient defenses against HTTP GET attacks, which is yet another reason for their popularity among cyber criminals. Defense methods are an active area of research in the academic and industrial cyber security community. The current portfolio of proposed solutions requires significant resources in the form of historical state information, complex analysis methods, deep packet inspections, special-purpose hardware (e.g., FPGAs), etc. Most, if not all, of these solutions are impractical and are not likely to transition out of the laboratory. There are a few products on the market, but they are expensive, complex to manage, not scalable and they adversely impact network performance. There is little to no published data on their effectiveness, efficiency and performance.

CNI has studied the available scientific literature on HTTP GET DoS attack solutions, while analyzing them for efficiency and simplicity. Our conclusion is that in general, the solutions do not focus on the discovering the most efficient methods for discriminating between bots and browsers. Unnecessary amounts of historical data is stored, analysis methods are unnecessarily complex, packet inspection is unnecessarily deep, the cost of processing resources is too high, performance and scalability are not properly considered and practical deployment considerations are not addressed. Hence, although an efficient, effective solution has not yet been delivered to the market, CNI believes such solutions exist.

Malware Attacks & Defenses

In a malware attack, a resource is tricked into executing applications that cause damage to, steal from, or otherwise misuse the resource. There are multiple vectors by which malware can intrude into an enterprise, such as infected media (memory sticks, disks, etc.), but the most popular and most efficient vector for adversaries to use is the Internet. Internet-borne malware attacks are generally considered to be an enterprise-only network security problem. ISPs are generally not expected to defend against malware attacks[13]. This discussion, therefore,

[12] The source IP address of packets containing HTTP GET requests cannot be spoofed because an HTTP session is transported using TCP. Before a bot or browser can send an HTTP GET request packet, a TCP connection with the web server needs to be established first, and to do this, the source IP address has to be correct.

[13] This view is not shared by Centripetal Networks, Inc. (CNI). CNI believes that ISPs can offer revenue generating malware defense services to their enterprise customers that significantly mitigate the effects.

examines the issue from an enterprise networking perspective. Malware comes in many forms, and for the purposes of this white paper, it is impractical to provide a comprehensive description of all of them. The most economically important class, however, is malware that commits ex-filtration – theft of sensitive data via the Internet – which has been identified as the number one cyber threat facing US networks. Accordingly, in this white paper, the malware description focuses on ex-filtration, conventional defenses against them and why they don't work.

Ex-filtration Attack Models & Conventional Defenses

Ex-filtrations are thefts of sensitive data via the Internet. Ex-filtrations are perpetrated by:

1. **Malware:** Malware is often surreptitiously downloaded onto a host that contains sensitive data (e.g., military secrets, financial account information, etc) or that is used by a human operator to generate sensitive data (e.g., credentials such as account login information and PIN codes). The malware finds or collects the sensitive data and sends it over the Internet to collection servers, without the host owners/operators being aware of the theft. Malware for ex-filtration often have a particular structure classifying them as trojans. Many famous ex-filtrators are trojans.

2. **Humans:** Human operators may either intentionally or unintentionally (e.g., via e-mail phishing attacks, etc) send sensitive data to collection servers.

Exploiting Stateful Firewalls

At first glance, one may naively assume that conventional network firewalls may be used to prevent ex-filtration. Since firewalls primarily control access to enterprise networks by filtering inbound packets, they should have some basic capability to detect and block packets that compose an ex-filtration. But, the behavior of conventional firewalls is easily exploited to perpetrate ex-filtration, as follows:

Firewalls readily block unsolicited packets that are destined for an enterprise's private resources (e.g., a desktop PC, data center, or the enterprise's network infrastructure) that are located behind the firewall, i.e., on the protected side of the enterprise network security boundary. A conventional firewall's trust model assumes that any session initiated by a resource located behind the firewall can be trusted. If the resource is actually malware, the firewall trusts it anyway because it can't distinguish between the types of resources that have initiated sessions. Thus, any inbound packets that have been solicited by a protected resource located behind the firewall – such as packets containing a web page requested by a web browser – are allowed to cross the security boundary.

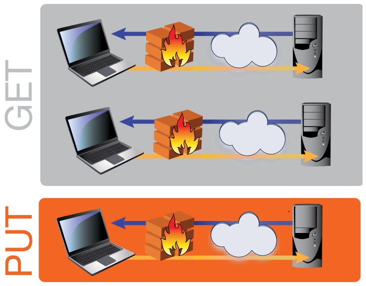
Firewalls implement this trust model by maintaining state information on Internet sessions initiated by resources located behind the firewall; accordingly, these firewalls are characterized as stateful firewalls. In a nutshell, stateful firewalls examine outbound packets and record the source and destination IP address and the source and destination port (values located in the packets' headers). Then, an inbound packet with source/destination IP addresses and ports that match an outbound packet's destination/source IP addresses and ports is allowed to cross the boundary.

This trust model and stateful firewalling are exploited by trojans and phishing attacks to perpetrate ex-filtration. Ex-filtrations by trojans[14] are typically perpetrated as illustrated above in Figure 1 and as described below:

Step 1: A link to a malware server is inserted, via infection, into a web page of an otherwise-legitimate web server.

Step 2: The user who downloads the web page is enticed to click on the malware link. This initiates a download from the malware server; the firewall assumes insider initiated communications are trusted, so it allows the download. The malware is installed on the user's host machine, and begins collecting sensitive data (e.g., usernames and passwords to user's online bank accounts) or locating locally-stored files containing sensitive data.

[14] R. Van Antwerp, "Ex-filtration Techniques: An Examination and Emulation", University of Delaware Library, 2011.



1. Victim surfs to legitimate web site that has been infected. Web page has hidden link to malware server. Firewall/ IPS bypassed using Assumption of Insider Trust.

2. Victim enticed to click link to malware server. URL blacklist system does not work since hidden link is IP address vs. domain name. Firewall/IPS bypassed using Assumption of Insider Trust. Web proxy bypassed by using non-standard port or CONNECT tunnel and/or encryption. IDS/IPS bypassed using encryption or polymorphic malware.

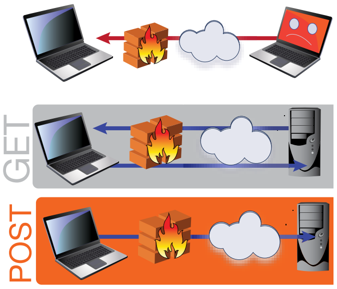
3. Malware finds/creates sensitive data files on Victim's machine. Malware sends sensitive data to collection server. Firewall bypassed using Assumption of Insider Trust. Web proxies bypassed by using non-standard Port or CONNECT tunnel and/or encryption. IDS/IPS bypassed using encryption or polymorphic malware.

Figure 1: Life Cycle of a Trojan Malware Ex-filtration

Step 3: The malware then ex-filtrates the data and/or files by sending them to a collection server. Again, the firewall allows the ex-filtration since the malware appears to be a trusted insider. Some malware even encrypts the ex-filtration content (using, e.g., TLS) to avoid detection by deep-packet-inspection (DPI) systems, such as IDS/IPS solutions.

Phishing attacks/ex-filtrations are similar in structure to trojan malware attacks/ex-filtrations. In a typical spear phishing attack (illustrated below in Figure 2), the victim is enticed to open an e-mail attachment – which actually launches a malware application – or is enticed to click on a link in an e-mail which appears to be a legitimate request from another user known to the victim, or from a business organization, e.g., a bank, known to the victim. The link takes the victim to a web site that either downloads malware or requests sensitive information from the victim (e.g., bank account login credentials). In any case, the last step in a phishing attack is the same as Step 3 of the trojan attack described above.

Finally, a human user, aka an insider attack, may perpetrate an ex-filtration deliberately. A human user will use some data transfer mechanism, e.g., posting files on a web site, instant messaging attachments, e-mail attachments, etc., to transfer sensitive data to a collection site.



1. E-mail sent by attacker to phishing victim. E-mail includes web link to malicious web site, which appears to be legitimate. Firewall/IPS not applied to email.
2. Victim enticed to click on link, which brings victim to "legitimate" web site familiar to victim (e.g. online banking). Victim enters SSN, username/password, etc. into a web form. Firewall/IPS bypassed using Assumption of Insider Trust.
3. Victim hits "Enter" (posts the web form). Firewall/IPS allows POST using Assumption of Insider Trust. Attacker now has valuable data to sell or exploit.

Figure 2: Life Cycle of an E-mail Phishing Ex-filtration Attack

Evolving Malware, Encryption, and DPI

Traditional firewalls filter on the so-called 5-tuple packet header fields: source and destination IP address, source and destination port, and protocol type (IPv4) or next header (IPv6). Traditional firewalls were effective when (a) the sources of cyber attacks were few, thereby limiting in practice the number of filter rules to a few hundred, or at most a few thousand, rules; and when (b) applications only used the well-known ports (e.g., port 80 for HTTP) standardized by IANA. As the cyber threat grew in size and sophistication, however, the number of filter rules necessary to provide effective protections also grew past the ability of firewalls to apply them while maintaining sufficient network performance. For example, firewalls and router ACLs typically enforce network security policies composed of hundreds, thousands or (infrequently) tens of thousands of rules; whereas, at any given time there are several hundreds of thousands, or even a few million, known bad IP addresses which should be filtered to provide networks with comprehensive protections from Internet attacks. Also, since there is no capability in the TCP/IP protocols nor in host operating systems to enforce standard port usage, malware can easily subvert the firewall port filtering rules.

Attackers have responded to DPI methods in at least three ways:

1. Diversity: Continually creating new malware and attacks. As of March 2011, more than 10 million malware signatures had been identified, with many more unidentified malware believed to be in existence (estimates as high as 200 million have been reported). DPI-based systems applying signature matching cannot scale to the size of the threat;

2. Adaptability: Malware and associated attacks are continually modified so that yesterday’s signature databases no longer match today’s attacks and malware; and

3. Encryption: Ex-filtration sessions may be encrypted, thereby thwarting DPI-based ex-filtration prevention methods that examine content. The popularity and ubiquity of the Transport Layer Security (TLS) protocol – which puts the “S” in HTTPS – for encrypting packet content has made it simple and cheap for attackers to conceal ex-filtration.

Malware diversity/size, adaptability (polymorphism), and encryption have made DPI-based ex-filtration prevention a la IDS/IPS extremely inefficient – essentially ineffective.

Figure 3 is a notional chart produced by the US Office of Naval Research (ONR)[15] in 2010. It shows the rapid divergence over time, of malware capability vs. malware defense capability. As stated in the chart, ONR declares that conventional malware defenses – anti-virus methods (intrusion prevention) and firewalls – are **ineffective.**

Besides the intrinsic difficulty of preventing malware intrusions, there is significant concern that even if IDS/IPS based defenses were effective, they cannot be effectively deployed at sufficient scale to protect enterprises against Internet-borne attacks[16]. This concern was made clear in recent US Government project called EINSTEIN 3. The program surfaced significant technical concerns as well as policy concerns. Areas of technical concern include scale, fast correlation ability (quickly recognizing new threats), device management, signature management and the need for man-in-the-middle decryption. Areas of policy concern include what to do about encrypted traffic, privacy issues, massive data storage requirements, potential misuse and abuse of the system’s devices and cost.

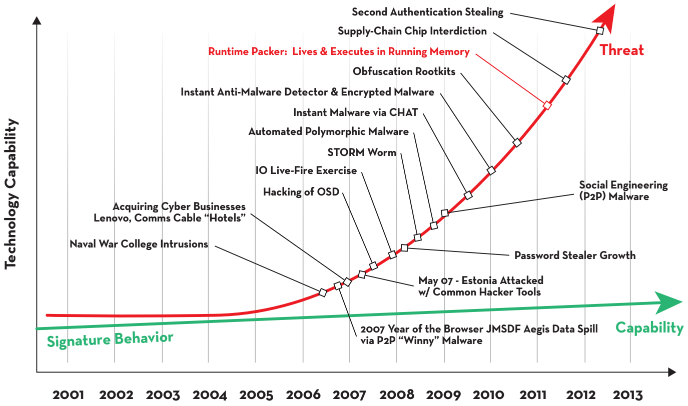


Figure 3: Notional chart from the Office of Naval Research (ONR) showing explosive growth of malware capability vs. linear growth of malware defense capability.

New Strategies in Ex-filtration Malware Defense

It appears that the cyber security industry is coming to the realization that although there is certainly some value in intrusion prevention as a malware defense strategy, it is not the panacea that the industry once hoped it would be. Thought leaders are now promoting a strategy that assumes malware infection cannot be totally prevented, and therefore emphasis should be placed on mitigating or neutralizing malware effects.

In the case of ex-filtration, this defense strategy is called ex-filtration prevention. An ex-filtration prevention strategy is realized by making it difficult or impossible for malware to transmit stolen data over the Internet. Clearly, conventional enterprise cyber security technologies – firewalls and IDS/IPS solutions – cannot help. Firewalls’ (false) assumption of insider trust offers no defense at all for ex-filtration – in fact, firewalls are the key enabler for ex-filtration, as described above. IDS/IPS solutions, by definition, attempt to solve the converse problem of intrusion prevention and therefore will not be part of an ex-filtration prevention solution.

Thus, it appears that new cyber security strategies and technologies are necessary to prevent ex-filtration. To that end, emerging ex-filtration prevention research has been focused on behavior analysis. The concept is to be able to characterize normal, legitimate behavior of hosts, and then be able to detect ex-filtration – in real-time – as anomalous behavior. This approach is analogous to the signature analysis approach used in intrusion prevention,

[15] S. Chincheck, “Computer Network Defense/Information Assurance (CND/IA) Enabling Capability”, BAA 10-004 Computer Network Defense, Industry Day, US Naval Research Laboratory, 24-Feb-2010, Washington DC. Available at http://www.onr.navy.mil/~/media/Files/Funding-Announcements/BAA/10-004%20Industry%20Day%20Overview%202-24-10.ashx

[16] S. Bellovin et al., “Can It Really Work? Problems with Extending EINSTEIN 3 to Critical Infrastructure”, Harvard National Security Journal, Vol. 3 (2011), pps. 1-38.

and it may be as difficult and complex. A recent thesis on ex-filtration methods demonstrates shows that the data transfer protocols and methods used by ex filtration are nearly identical to those used by legitimate functions[17]. If behavior analysis proves to be as complex, if not more so, as signature analysis, then this cyber security strategy may also fail to solve the problem. If this is the case, then hopefully it will be realized early on so that R&D can be focused on more promising strategies and technologies.

Why Cyber Security Isn't Working – and What Can Be Done About It

The evidence shows that despite huge investments in cyber security, it is not working. Thousands of successful attacks are launched every day. The problem continues to be highly asymmetric; a small number of adversaries using cheap tools can inflict tremendous damage and loss, even on those networks using the most sophisticated (and expensive) cyber defenses. CNI believes cyber asymmetry can be reversed. The preceding analysis of Internet attacks and conventional cyber defense methods reveals potential paths to explore.

For the case of DoS attacks, one fundamental problem is that conventional packet filtering technologies – network firewalls and router ACLs – do not scale to the size of the threat, nor can they adapt rapidly enough to track with changes in the threat. If scalable, adaptable and cost efficient packet filtering technology were available, then it could be highly effective in mitigating DoS attacks. Such a breakthrough technology would reverse the asymmetry, making it relatively simple and cheap for networks to defend against DoS attacks, while making it much more difficult and expensive for adversaries to launch successful DoS attacks.

For the case of malware, and in particular ex-filtrating malware, one fundamental problem is that an ineffective strategy has been pursued. Until recently, the defense strategy has been to prevent malware intrusions via the network. The evidence shows that not only has this intrusion prevention strategy failed, but also during the time the strategy has been pursued, the asymmetry has become overwhelmingly skewed to the advantage of the adversaries. An important note here is that in the case of ex-filtration, there is no harm done when malware intrudes into a network. Instead, the harm is done when the malware ex-filtrates valuable information. Hence, changing the strategy to one of ex-filtration prevention will provide real cyber security benefits. Surprisingly, however, it appears that the cyber security industry is only reluctantly moving towards an ex-filtration prevention strategy. Moreover, the current research focus on behavioral analysis is not likely to be effective, let alone to reverse the asymmetry.

A better approach to ex-filtration prevention may be to establish and enforce network communications policies that specify which resources are allowed to communicate with each other, and how they can communicate with each other (e.g., read-only, read-write, (un)encrypted, etc.). One way to enforce such policies is packet filtering. The bad news, however, is that packet filtering technology must become much more scalable, adaptable and agile if it is to solve the ex-filtration problem. But the good news is this is the same obstacle faced by packet filtering in solving the DoS attack problem. Hence, if a breakthrough in packet filtering technology is made, then the two biggest cyber security problems – defending against DoS attacks and preventing ex-filtration – can be addressed simultaneously, and the asymmetry can be reversed.

[17] R. Van Antwerp, "Ex-filtration Techniques: An Examination and Emulation", University of Delaware Library, 2011.