Network Performance and Security

Testing and Analyzing Using Open Source and Low-Cost Tools

Chris Chapman

Steve Furnell, Technical Editor

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Syngress is an imprint of Elsevier

SYNGRESS

Syngress is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, USA

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-803584-9

Publisher: Todd Green
Acquisition Editor: Brian Romer
Editorial Project Manager: Anna Valutkevich
Production Project Manager: Punithavathy Govindaradjane
Designer: Matthew Limbert

Typeset by Thomson Digital

This book is dedicated to Joan. Without her,
nothing would be possible.

CHAPTER

1 Introduction to practical security and performance testing

This book is intended to help you practically implement real-world security and optimize performance in your network. Network security and performance is becoming one of the major challenges to the modern information technology (IT) infrastructure. Practical, layered implementation of security policies is critical to the continued function of the organization. I think not a week goes by where we do not hear about data theft, hacking, or loss of sensitive data. If you dig deeper into what actually happens with security breaches, what you read in the news is only a small fraction of the true global threat of inadequate or poorly executed security. One thing that we all hear when an article or a news item is released is excessive amounts of buzz words around security, with little content about how it may have been prevented. The truth is, security mitigation is still in its infant stages, following a very predictable pattern of maturity like other network-based technologies. Performance is another critical part of a well-performing network. Everyone knows they need it, but to test it and measure it is not only a science, but also an art.

I assume that the reader of this book has a desire to learn about practical security techniques, but does not have a degree in cyber security. I assume as a prerequisite to implementing the concepts in this book, the reader has a basic understanding of IT implementation, has a mid level experience with Windows and Active directory, and has had some experience with Linux. Furthermore, my intent in this book is to minimize theory and maximize real-world, practical examples of how you can use readily available open source tools that are free, or relatively low cost, to help harden your network to attacks and test your network for key performance roadblocks before and during deployment in a production network. In fact, the major portion of theory that I will cover is in this chapter, and the focus of that information will be on giving you a baseline understanding in practical deployment and applications of security and performance. I also assume noting, and will take you through execution of best practices.

A BASELINE UNDERSTANDING OF SECURITY CONCEPTS

What is an attack? It is an attempt to gather information about your organization or an attempt to disrupt the normal working operations of your company (both may be considered malicious and generally criminal). Attacks have all the aspects of regular crime, just oriented toward digital resources, namely your network and its data. A threat uses some inefficiency, bug, hole, or condition in the network for some specific

objective. The threat risk to your network is generally in proportion to the value or impact of the data in your network, or the disruption of your services no longer functioning. Let me give a few examples to clarify this point. If your company processed a high volume of credit card transactions (say you were an e-commerce business) then the data stored in your network (credit card numbers, customer data, etc.) is a high target value for theft because the relative reward for the criminals is high. (For example, credit card theft in 2014 was as high as $8.6B [source: http://www.heritage.org/research/reports/2014/10/cyber-attacks-on-us-companies-in-2014].) Or, if your business handles very sensitive data, such as patient medical record (which generally have the patient-specific government issued IDs such as social security numbers attached), you are a prime target. In either case, the value of data in your network warrants the investment and risk of stealing it. Say, you are a key logistics shipping company, the value to the attacker may be to disrupt your business, causing wider economic impact (classic pattern for state-sponsored cyber terrorism [example: http://securityaffairs.co/wordpress/18294/security/fireeye-nation-state-driven-cyber-attacks.html]). On the other hand, if you host a personal information blog, it is unlikely that cyber crime will be an issue. To put it bluntly, it is not worth the effort for the attackers. The one variable in all of this is the people who attack network "because they can." They tend to use open source exploit tools, and tend to be individuals or very small groups, but can be anywhere on the Internet. We have to be aware of the relative value of our data, and plan security appropriately.

There are many ways of attacking a network, let us spend a few moments and cover some of the basics of security and performance. If we divide attacks into their classification, we can see the spread of class of attacks growing over time. What types of attacks may you experience in the production network?

DDoS ATTACK

DDoS, or distributed denial of service, attacks are an attack class with the intent to disrupt some element of your network by utilizing some flaw in a protocols stack (eg, on a firewall), or a poorly written security policy. The distributivenes comes into play because these attacks can first affect devices such as personal computer (PC) or mobile device on the Internet, and then at a coordinated time, can attack the intended target. An example would be a TCP SYN flood, where many attempted, but partial, TCP connections are opened with the attempt to crash a service on the target. DDoS attacks may also be blended with other exploits in multistage attacks for some multistage purpose.

BOTNET/WORM/VIRUS ATTACK

A botnet is a code that first attempts to install its self within the trusted portion of your network, though combined and blended attacks may spread to other resources across your network. A botnet has two possible objectives. First, spread as far and as fast as it can within the target domain and then at a specified time, bring down elements in the network (like PCs). Second, a botnet can quietly sit in the network, collect data, and

"phone home" back to a predefined collection site over well-known protocols. This is considered a scrapping attack because data are collected from behind your firewall and sent over known-good protocols such as HTTP/HTTP(S) back home.

TROJAN HORSE

A trojan horse is a type of attack that embeds the malicious code in some other software that seems harmless. The intent is to get the user to download, install, and run the innocent software, which then will case the code to infect the local resource. Another great example of this is infected content that is downloaded off of P2P networks such as Bittorent; the user runs the content and the malicious code is installed.

ZERO-DAY ATTACK

A zero-day attack is a traffic pattern of interest that in general has no matching patterns in malware or attack detection elements in the network. All new attacks are characterized initially as zero-day attacks.

KEYLOGGERS

A keylogger is a code that is installed by malware and sets on a device that has keyboard input (like a PC) and records keystrokes. The hope of the keylogger is that it will capture user login credentials, credit card number, government ID numbers, which can later be sold or used. Keylogger can be deployed by botnets, or themselves be deployed. Variants of keyloggers will look at other inputs and records. For example, variant code may listen to your built-in microphone or record video from the integrated camera (or just take periodic snapshots).

SQL INJECTION ATTACK

Chances are you have an SQL database somewhere in your network. Attackers know this and know by its very nature that the database holds valuable data, or at the least is a choke point in the workflow of your business. An SQL injection attack uses malformed SQL queries to perform one of two possible functions. First, the simplest attack is to crash some or part of the database server. This has the obvious effect of stopping business workflows. Second, an SQL attack may be used to selectively knock down part of the SQL server, exposing the tables of data for illicit data mining.

CROSS-SITE SCRIPTING ATTACK (XSS ATTACK)

The modern platform for application is the web. What this means is that the sophistication of what is served and processed has greatly increased. The web has moved from a simple text-based system to a full application API. A cross-site scripting attack takes advantage of this sophistication by attempting to modify the middle ware of the web application. For example, it may insert JavaScript inside of code to bypass

a login, capture data, and phone home or become purely malicious. This class of attack is a good example of how attackers desire malicious code to be undetected for as long as possible, especially when the exploit is attempting to collect data.

PHISHING ATTACK

A phishing attack can come in many forms, but generally focus on web content modification and emails. The idea behind a phishing attack is to look legitimate, attempt the target to give sensitive data, and capture/sell the data for profit or use it for malicious means.

ROOTKIT

A rootkit is a special type of worm that can embed its self deeply into the operating system (thus the "Root") such that it can take over the system involuntarily. Rootkits can be very difficult to remove and detect.

FIRMWARE VIRUS

A firmware virus will attempt to reflash elements that have firmware, such as your hard drive or PC EFI. This is related to the rootkit family of attacks and in some cases can physically destroy equipment. For example, a virus inserted in a hard drive firmware can destroy the lower layer formatting of the drive, or corrupt TRIM setting to accessibly use SSD memory cells to failure. On a server, EFI virus could increase CPU core voltage and turn off fans to cause death by heat.

HIJACK ATTACK/RANSOMWARE

This class of attack attempts to take a legitimate active session and insert or redirect data to a collector. For example, imagine an e-commerce session, where users shipping and credit card information is captured. This class of attack is sometimes called a "Man in the Middle" attack. In the case of Ransomware, the attack will shut down the device functions and make the user pay, sometimes even a small amount, to "unlock" their PC. Attackers know that if a user pays, say $5, to "recover" their gear, it may not be worth reporting. This, multiplied by millions, can be big business.

SPOOF/EVASION ATTACK

In this class of attack, the attacker intentionally rewrites Ipv4, UDP, and TCP fields to try to hide from firewall rules. For example, if I take an attack and use IPv4 fragmentation, I might be able to hide the attack from the firewall policy rules, because as the attacker, I hope the firewall pattern matching code does not cover this condition.

BUFFER OVERFLOW ATTACK

Typically, network application, protocol stacks, buffers, and queues expect data request in a structured format. A buffer overflow attack will attempt to intentionally

send malformed or excessive data to "crash" some or part of the application, firewall, or any network element in between. Sometimes, this is called a knockdown attack.

PASSWORD ATTACK

This kind of attack uses automation to break a password by many iterations. There are three types of approaches: Brute-force, dictionary, and hybrid attempts. This is always a roll of the dice, but in some cases, especially with a dictionary technique, attackers know users have poor password selection habits, and will try clusters of known combinations first.

PENETRATION ATTACKS

A penetration attack is more complicated than other types of attacks, because it tends to be multistage, distributed, and orchestrated. These types of attacks can be the most damaging, because generally they require a level of sophistication and resources to achieve their target. Many security breaches you might hear about in the news are sophisticated penetration attacks, especially if there is a large volume of data theft. Penetration attacks are like high stakes poker. It requires skills, patience, strategy, and stages, but has very large payouts if successful.

MALWARE

Malware is a generic class of attack that may refer to distributed as trojans, worms, botnets via applications, websites, or emails. Malware is the most prodigious form of attacks, with Q4 millions of variants flowing through the Internet annually. It should be noted that attacks can form hierarchies. For example, malware may be used to insert rootkits or keyloggers. Malware may also insert other malware as a cascading infection through your network.

VOLUMETRIC ATTACKS AND ATTACK FREQUENCY ACROSS THE INTERNET

With over 82,000 new malware attacks daily [source: http://www.pcworld.com/article/2109210/report-average-of-82-000-new-malware-threats-per-day-in-2013.html], it should be assumed that you will be attacked hourly. It is projected that by 2020, this rate will increase to over 100 GBps per day, every day, 365 days a year [source: https://www.neustar.biz/resources/whitepapers/ddos-protection/2014-annual-ddos-attacks-and-impact-report.pdf]. So form the perspective of your network; it is a safe bet that each and every day you will be either directly targeted or indirectly experience attacks on your public Internet peering points. Understanding this point is very important, because it is no longer "when" but "how and where" you will be targeted. Knowing that you will be perpetually attacked, and still having

the requirement of transacting business over the Internet is a critical mindset toward security and performance of the modern network.

There are two really good websites that will show live attacks based on a world map.

NorseIP (http://map.ipviking.com/) and Digital Attack Map (http://www.digitalattackmap.com/#anim=1&color=0&country=ALL&list=0&time=16843&view=map) will show live attacks based on country.

Both of these sites should be used to see patterns of attacks across the Internet. The intent is to demonstrate scope and scale of attacks that happen daily.

SECURITY NETWORK ELEMENTS

So, what are the devices and subsystems in the network that can help manage security? These devices should always allow with minimal impact for valid user workflows while catching and mitigating attacks.

Here are some of the devices.

DISTRIBUTED FIREWALL

Original firewalls were a single appliance with a trusted, untrusted, and DMZ network connections. They would have a policy that would allow or drop conversations. This model has evolved into a distributed firewall, which will allow you to write an enterprise-wide policy and distribute it across key peering points in the network as well as firewall nodes sharing threat information network wide. So what are some of the functions of the modern firewall.

Traffic access policy

Access policies are the rules that you decide you wish to allow between zones. Implicit in these policies is a "Deny All" which is implied at the end of your policy. Therefore, traffic you explicitly do not allow should be denied. The concept of creating the smallest number of "Pinholes" in the policy is considered a best practice. Older firewall technology was based on ACCPT/DENY/IGNORE rules on the basis of destination TCP or UDP port numbers. This class of firewall is considered obsolete and should not be used. The reason is simple, destination port numbers are far easy to spoof. The modern policy will not only know the transport protocols like HTTP, but it should also understand services such as specific web applications and SIP.

Access control

Access control is a Go/No-Go policy that looks at source, destination, and traffic and makes a decision to allow or deny a conversation. It is considered "mild" security, but is useful to deploy in a layered security model.

Location management

Where is the user geographically sourced. Are they from an approved location or not?

User management
Who specifically is using the application, and are they authorized?

Access times
Is this user allowed this workflow at this time, or not?

Workflow intelligence
Is this person allowed access to this par of the application or not?

Logging
Logging, or documentation of event to a central logging server, will keep a historical record of events. Logging can be very CPU intensive, so what and how you log is critical. Best practice is to log negative events. In some jurisdictions, logging is becoming a legal requirement.

Remote access/VPN
Remote access, generally subdivided into site-to-site (remote branch) and remote access (point-to-point) virtual private network (VPN), is a technology that creates a tunnel through the Internet that is secure and encrypted. The main flavors are IPSec (older) and SSL-VPN (newer).

IPS/IDS
The purpose of this element is to detect or prevent intrusion and perform some action. Typically, this element will either be passively inline with traffic (IPS) to allow it to block attack, or hang off of a network tap (IDS) such that the element will detect and perform some action. For example, the IPS/IDS service contains a database of patterns that predict an attack. If a traffic flowing through the appliance triggers three patterns, and IDS will log the event, IPS will attempt to block the traffic.

PROXY SERVER
A proxy server is a device that will terminate TCP connections and regenerate them on the outbound side. Typically, the user must configure the proxy server and port number in the local application, such as the web browser. Proxy servers can be a layer of protection, because they isolate traffic above TCP/UDP from the original connections. This has the benefit of potentially blocking TCP-based attacks. Proxy server should be considered a layer of security, but should never be deployed as the exclusive element of security.

TOR NETWORK
“The Onion Router” (ToR) is an anonymity routing technology that hides the identity of users through random path routing. A ToR shim is useful to evade specific pathways (where people may be spying) since it picks paths randomly. ToR is not absolutely secure, and must always be combined with other encryption to improve security.

PERSONAL FIREWALL/ANTI-VIRUS/ANTI-MALWARE

This class of security object is typically installed on the desktop. They tend to perform "Leaf" analysis, inspecting the local file system and memory for infections. They can use significant local resources, and generally require "syncing" to keep the local database up to date. The implication of the personal firewall is two fold. First, the firewall is only as good as the underlying technology used to scan traffic. It is possible for a firewall to miss an attack because the scanning engine was not engineered to detect the attack. Second, a firewall is only as good as its last database sync. The implication is that periodic work is required for all nodes to keep up to date.

A BASELINE UNDERSTANDING OF NETWORK PERFORMANCE CONCEPTS

Network performance is one of those topics that everyone knows they need, but is hard to quantify and qualify. I would like to spend a moment and discuss network performance, what it is, what effects performance, and how it is measured. Network performance is related to security in the sense that both attack mitigation, attack effect and quality of experience must be measured together. All three vectors of performance are considered entangled.

WHAT IS NETWORK PERFORMANCE?

First, network performance is not one thing; it is many things working together for a user experience. As more services are placed on the network such as voice, video, data workflows within the company, a decrease in user experience is not only annoying to the users but can sharply reduce productivity in the network. So, performance testing is really the study of user experience over time in the network from different populations, under different conditions.

PSYCHOLOGY OF USER EXPERIENCE

The modern user relies upon the network being available, always on, predictable, and reliable. For example, if a user browses to an internal CRM system, and gets a "*404 Error*," the event breaks availability and predictability requirements of the user, and perceptual experience goes down because they simply cannot do their job. In many ways, the old POTS (Plain Old Telephone System, or pre Internet voice networks) dial tone concept of 99.999% uptime at fiber optic quality (remember the old add of hearing a pin drop) has set a benchmark and is transformed into a form of a Web Dial tone expectation. This is simply a reflection of the reliance of the user upon the network for even fundamental day-to-day operations. The first attribute of how users perceive user experience is that they do not recognize when workflows perform well; they simply expect this day in and day out. This is an important attribute to recognize,

because the network can have a great experience factor for a full year, and users will tend to not take that fact into consideration. When a user in the network perceives a negative event such as a slow loading page, or disrupted voice quality on SIP, then they place a very strong weight on the times the network did not work vs. the times it did work. In general, users just expect the network and its services to just work all the time. Furthermore, users frame their experience on the basis of the service, not the protocol. What I mean by this is they will see the "CRM" is good or bad, not HTTP and user experience is a measure perceived impairment for workflow within the service. So what can go wrong in a service? These are divided into hard and soft errors.

HARD ERRORS

When a user cannot login (authentication problem) or receives a "404 page not found" error a hard error occurs. These events can occur randomly, periodically, or one shot. They are very measurable and discrete because the condition either exists or does not exist. Hard errors have a lot of perceived weight by the user because it directly prevents them from completing their task, increasing frustration. In addition, a hard error can be weighed on the basis of when and where it occurs. For example, if a user cannot log in to the CRM, the hard error impact on user experience can range from annoying (low impact coefficient) to panic (high impact coefficient) on the basis of the specific user condition and criticality of the desired user action. The bottom line on hard errors is that they are never good, they contribute the greatest to a negative user experience, they can be perceptually multiplied based on the user situation, and they take a very long time to balance out with well-performing workflows.

SOFT ERRORS

If a hard error is black and white, a soft error is a shade of gray. They tend to be expressed as slowdown of a service which can occur randomly, periodic, or persistently. The tendency of the user to notice a soft error is directly proportional to critical nature of the service and where the user is within the workflow. Soft errors impact the perception of quality in a meaningful but different way than more direct hard errors. Whereas a hard error such as a page not found is perceived as a definite failure, soft errors like slow loading pages, or high variability in page time loading will cumulatively degrade the perception of quality over time. Users will assign more negative impact to soft errors on the basis of frequency, cluster events, or if there is a perceived pattern of slowdowns. For example, if a user between 8 and 9 am each day sees the CRM system to be "slow" they will place much more negative influence, such as a hard error coefficient, than if it happened "last Tuesday, one time," which tend to be more easily dismissed by the user. Users recognize patterns, and give extra weight to those patterns.

Hard errors are remembered for long durations of time, especially if there is a high coefficient of effect. This is then followed by pattern or clusters of soft errors, followed by nonreoccurring soft errors. These experience events do also get

examined by the users as a set. So seeing periodic hard errors and clusters of soft errors dramatically lowers user experience.

QUALITY OF EXPERIENCE FOR WEB-BASED SERVICES

So much of our crucial work is done by web applications. The trend is for the web to take over desktop applications, and effectively become a programmable interface. Web-based services, such as CRM, order processing/logistics, or even a replacement for office applications, go straight to the core faction of the company. Understanding how to measure web services is crucial to understanding not only good user experience, but also the impact of security events on user experience. So what are the attributes of a web-based service, and how do users perceive quality.

The web service may use web technologies, but is by no means a classic static web page. A service is a fully stateful application, spilt between the web browser and the back end. As such, the applications differentiate users through some form of authentication. Several hard error events may occur. If the user browses to the login page and receives a 4xx error or they type in valid credentials and the web application freezes, the user will defiantly perceive this as a hard error. Authentication may also experience soft errors. In general, an application is considered to have excellent authentication performance if the user can be authenticated and logged in within 1 second or less, 2–3 seconds is considered acceptable and >4 seconds is considered to be "slow."

Web-based application tends to have screens and workflows through screens to perform actions. At the individual screen level, if the page does not load or if an object on the page is missing, then the screen is experiencing a hard error. As a rule of thumb, if the page renders in <1 second, the perception is excellent, >2 seconds, and the page begins to be perceived as slow. At >7 seconds, most users will simply abort, and consider the page to have experienced a hard error. Many pages also require the user to post data to the server; the same timeframes for page load time are reflected to a post response page from the server.

So a test workflow would have a user authenticating and walking through workflows that are considered typical and critical to the organization. The test is looking out for hard error events and measuring authentication, page render time, and data post times for the application.

QUALITY OF EXPERIENCE FOR VOICE

Voice on the network is a critical service that most people rely on day after day. The measurement of quality of experience is more defined for voice using an MOS algorithm based on ITU P.861 and P.862. This score is a range from 1 (unacceptable) to 5 (excellent). In general, you want a score of 4.2 or higher.

> *Why does MOS (Mean Opinion Score) matter? MOS scoring was derived by gathering a statically significant number of callers, and having them rate the call quality across numerous sample (Source: http://voip.about.com/od/voipbasics/a/MOS.htm). It was assumed that person would use the pre-Internet fiber optic*

landline phone network as a benchmark. MOS scoring was derived from this study. It is assumed that MOS scoring is a factual and meaningful measure model that predicts how users will judge voice quality. Given that we use the phone daily, it should be considered a core service in the network.

We have to differentiate what we are measuring, an MOS score will measure an impact of the network on voice, but that does not translate into excellent call quality. Say, for example, the handsets have a bug or simple do not decode voice well, no amount of network tuning will get you acceptable quality. It is strongly recommended that you ask the handset and IP phone vendor to specifically test and demonstrate SIP through their device. In addition, hard errors can also occur in specific SIP functions such as bridging, call transfer, voicemail, etc. We will not cover these specific types of hard error events, but you should be aware of them.

QUALITY OF EXPERIENCE IN VIDEO

Video in the network is going through a rapid transition from older format (Flash/RTMP) to a more modern ABR HTML5-based video. The big difference is how video is encoded and transported. This book will exclusively deal with ABR HTML5 video around the DASH standard, which is the most relevant in the network. ABR video is video transported over HTTP protocol. We must also make the same declaration about understanding what domain is tested as voice. We will focus on the impact of the network on video, but we will not cover encoder/decoder quality. It is mentioned here because the encoder and decoder can make a well-transported video stream look good or bad. HTML5 video performance is measured in HTTP goodput as a ratio of offered versus received bandwidth. HTTP goodput is the data rate between the HTTP stack and the video decoder, so it takes into consideration TCP events, network conditions, etc. In addition, as a ratio normalized as a percent, the user can easily measure even the most modern technologies such as 4k video streams. For an excellent video, the user wants a goodput ratio of 98%+, consistently through the video stream.

NETWORK EVENTS THAT CAN EFFECT HARD AND SOFT ERRORS FOR FLOWS

A flow is a pathway across the network connecting a client and a server, whereby data flow through the path. That pathway can become impaired, and it is important to understand how different types of impairments effect user experience. Here is a list of some common network situations that can reduce user experience.

BANDWIDTH CONSTRICTION

Typically, there are many hops in the network between the client and the server. Bandwidth constriction can happen anywhere in this chain, and tends to be a “weakest

link" event. Constriction of a flows bandwidth may either be based on a policy or a limit of a network element or elements. When the bandwidth pathway is restricted, TCP window size may never grow to MSS, and you will see artificially slower performance in such things as total page render time.

NETWORK LATENCY

Latency can play a big part in performance. Latency in a datacenter should be very small (100's of uSeconds). Across a WAN, on a point-to-point link, the natural "in the ground" latency is approximately 1 mSec per 100 km of distance. The effect of latency is that it can slow bandwidth and if latency is too high, TCP may time out, reduce the window size, and try to recover, which is expensive to performance. Too much latency can also effect audio quality, forcing a lower quality coded to be negotiated.

JITTER

This impairment is dynamic variation in latency across time. SIP stacks especially do not like jitter. This impairment is impossible to eliminate, but should be managed and capped at less than ±0.5% of the average latency, maximum.

CONVERSATION SEQUENCING ERRORS

The order of packets in a flow can make a big difference on the quality of experience (QoE); sequencing errors are a single or compound set of incorrect alignment of packet order. The impact of the sequencing errors depends upon the layer 4 (TCP or UDP) of the upper layer service. For TCP, there is generally a local buffer whereby if the packet sequencing error event resolves itself within the buffer, TCP can locally rearrange the packets to minimize the event. If the sequencing event is outside the buffer, it is treated like a lost packet event, and TCP will attempt to recover, eating up performance cycles. UDP-based services in the presence of sequence numbers have no local Layer 4 buffer, and rely on the upper layer service to recover. Thus, TCP is better than UDP in performance if the sequence event can resolve its self in the local buffer. The first kind of sequencing error is a lost packet. In this event the packet is simply dropped in the network. This will always force a recovery mechanism in either TCP or UDP's upper layer service. This is a very hard error, and never beneficial to the flow. The next kind of error is a reorder event, such as packets being swapped. As long as the event occurs within a short set of packets, this event is annoying but recoverable. Duplicate packets can have more of an impact, because for TCP-based services, the TCP stack has to recognize that it received something it previously received and discard it, which takes time and resources. Late packets look like lost packets, but eventually make it to the stack. Again, as long as this is within the local window, TCP can compensate (UDP cannot). Sequencing errors are never beneficial, and frankly you should not see them in a modern device.

SERVER CONGESTION

A primary cause of slow performance in a network is generally over subscribed servers or server pools. An oversubscribed server may save some money, but can cause a lot of performance problems in the network. A useful way of measuring server congestion is to load the server to near failure and then measure the time to first byte returned from the server. This metric measures after the 3-way TCP open, time it takes to receive the first byte of the first object. In scenarios where you have HTTP persistence and pipelining, you also look at object response time. Good response times should be in the milliseconds range (10's to 100's), a 500 ms response time would be questionable. Quantifying the server would mean that you load the server to the maximum number of users, and you look for maximum object response time across all users. If it is in the 50–200 ms range, and you have a reasonable sample set (ie, 600+ seconds of test data), then the server is correctly loaded. Another important attribute of the network is how resilience systems are for recover after an event, like a failure. In general, networks should converge in hundreds of milliseconds so that users do not experience the failure.

SUMMARY—BEFORE WE START TO HARDEN THE NETWORK

Network attacks and security planning are a critical component to the modern operations of a production network. It is safe to assume that your network will be attacked from both the outside and inside continuously. For this reason, we must build both external and internal defenses that balance the need to protect data and the full functioning of valid traffic, thus aggressively blocking unwanted traffic. In practicality, real-world issues in the network like a gap in inventory between what is thought to be in the network and what is actually in the network can reduce the overall reliability. In addition, we must always assume that the actual traffic flows in the network are different than what we think should be there. Because of this, we have to plan for the "unknown of unknowns." In addition, many networks are assumed to have a "trusted" (inside the network) and "untrusted" (outside the network) view of security. This all or nothing model of security actually lowers security because it does not consider that attacks may come from the "trusted" zones. Furthermore, the way the network is architected, like dumping VPN and Wi-Fi traffic straight into the core and not isolating them, adds pathways where malicious users or code gains easier access to critical areas of the network.

It is my experience that in practice, many organizations have poor documented and practiced emergency procedures. For example, a plan may be written down, but the format of the policy may be hundreds of pages long. In an emergency, an IT professional will not have time to read this level of volume. Even worse, there may be a disaster plan, but many organizations never "fire drill" and practice simulations of attacks. Another real-world issue is the reduction in security fidelity through "soft" trust events like password sharing.

> *Why is Trust bad for Security? Trust is a noncontrollable, nonmeasureable assumption that someone will always do the right thing. For example, a*

administrator may "Trust" a user with a admin password, trusting that they will only use it for a specific and limited use. How would you ever possibly know if this is what really happens? Could they not use the password for accessing other network resources, or add another admin level account? What about trusting that user will not bring in infected files (A USB stick for example). Trusting, or even educating user, not to do something is "Soft" security. It is nice if it works, but should never be a primary line of defense.

When we think of trust, we assume our vendors (hardware, software, cloud, services, etc.) are trustworthy. The truth is you should approach each vendor with skepticism. For example, Windows (7, 8, 8.1, and 10) will attempt of 30 different ways to "Phone Home" data [source: http://www.howtogeek.com/224616/30-ways-windows-10-phones-home/]. The vendor may claim this is for legitimate reasons. That may be true, that may also not be true. Who knows? Trusting vendors to perform all necessary security scans will always give you an insecure network. Primarily, vendors test generic scenarios and may focus on use case scenarios that are not relevant to your network. They only way you will know is to test yourself.

Last, we have to document the inside of the network. In the worst case scenario, the internal network is one large security zone (this is different from subnets, or how IP is routed). This is the best way for malware to spread and your network to be optimally infected. I have even seen in some cases that it was considered against the culture of a company to internally firewall. This is simply architectural foolishness. There is almost no justification of why every user in the network needs access to every resource. Our goal moving forward is to recommend how we can initially build or transform networks into more secure structures.

CHAPTER

Getting organized with initial audit of the network

2

Understanding that network elements, hosts, server, and wireless devices come in contact with your network is a critical first step in locking down the network. There are definable attributes that most networks staring out before a security and performance optimization exhibits. Most networks are not planned but instead evolve over time to their current state. In many cases, the network has passed ownership from different groups, staff members, or network architects. As a result, many overlapping priorities and generations of equipment exist on the network. The net result of this is that in many cases what is actually on the network, or touches the network via remote access or Wi-Fi, compared to what you think comprises the network can be very different. Even if you have been in control of the network, there may be legacy devices that exist, or place on the network without your knowledge. The impact of this on security can be profound. For example, if there is a "PC in the Lab" or "Private Server" somewhere that is not documented, then there is a high probability that security patches will not get applied, leaving room for such attacks as botnets. Once malware gets a beachhead into your interior network, it can be both extensive in terms of time and money to remove. Another problem of the "unknown of unknowns" of phantom devices is that you will never have positive identification of what belongs on the network and what does not belong on the network. Here, infections can reoccur even after a cleaned network. Furthermore, poorly planned Wi-Fi or virtual private networking (VPN) access may open up the core of the network to an almost unlimited number of "rouge" devices that are unknown, not secure, and will most certainly undercut any security policy we deploy. In addition, we must also point out the perils of virtualization. There is a very high chance that your network now uses hypervisors for their many benefits. The downside to a virtual server is simple that they are so easy to create and remove that you cannot visually inspect them like a physical server in the lab. Virtual servers can be just as dangerous as physical server, and in some cases more, especially if someone on the network is "experimenting" with software or Internet-based cloud services. In the age of BYOD (bring your own device—an organizational policy to allow employees to use their personal equipment on-site), having a well-planned security policy is very crucial. The bottom line is that if someone brings their own device, they may also bring their own malware, viruses, etc. This can have the obvious effect of making a secure network less secure. BYOD has many business benefits, so we are not saying do not allow it, but simply plan well for it and isolate it in the correct zone.

If we drill into the infrastructure, undocumented configurations and lack of limited, positive authority over configuration of those devices can led to disastrous effects. This is especially true when you think of a network device, such as Layer 3 switch, which may have a configuration such as an access control list (ACL). It may be legacy and may not conform to our planned security policy. So the first level effect of misinformation about how network elements are configured is lower security levels, but the secondary effect can be lower network performance. For example, if you have too many unnecessary ACL rules on a core router, you will slow down a majority of traffic on the network. For example, unoptimized or excessive lists may trigger a TCP timeout event. Most networks also employ some form of WAN. These can be in the form of VPN site-to-site IPSec tunnels, which is a predominant standard in creating encrypted tunnels across a public network or dedicated point-to-point service provider links. By not understanding basic information about service levels you may be paying for, or optimized link characteristics. The network may become sluggish, especially for remote users. Furthermore, understanding the basic domains of the network is a critical step in planning since we plan to use a system of zones to minimize threat exposure.

When we think of documenting the network, we seldom consider the data stored and transmitted through the network as an element of security. The truth is without understanding the sensitivity of information in the network, stored on hosts in the network, or transmitted across the network, we open ourselves to some of the worst possible data breaches. Remember that when the bad guys try to steal data, the operative word is "Data."

GOALS AND OBJECTIVES OF THIS CHAPTER: POSITIVE IDENTIFICATION OF VALID ASSETS

Now is the time to get organized and begin the process of hardening the network for security and performance. The first step I want you to do is take any network map, spreadsheet, or any other form of existing network documentation and throw it out. The reason why we do this is simple, stuff that is written down, without a chain of custody, even with a chain of custody, will in many cases be wrong or incomplete. It is important in our planning that we positively, personally identify each and every element on the network and not inherit and propagate bad or possibly incorrect information. We will start fresh and detect what is on the network. We will disregard trust, or what other people report, in favor of direct identification.

> Why not simply use network maps that currently exist? Existing maps, created by an arbitrary chain of administration over the years, may be right, but they may also be wrong. The problem is not knowing the varsity of the information. A well designed network that is secure and well tuned must begin with a very solid understanding of what is in the network and how it is configured. This helps reduce a false positive sense of security

In the first step, we will identify what host assets are on the network, which included fixed PCs, laptops, tablets, and smartphones. This process will use both automated inspection and quite a bit of manual inspection, followed by more automated differential inspection. By the end of this process, every device used to access the network will be positively documented. We will examine the make and model of the device, specific configuration data points like fixed IP address or Dynamic Host Configuration Protocol (DHCP), the MAC address of each and every device, the operating system and version, installed software (approved and not), what network management tools are installed, and specific device configuration and to whom the device is owned. From this information, we will populate a Network Management System (NMS) database.

In the second step, we will inspect and document server resources. We will specifically inspect server type (bare metal or virtual), server OS and patch level, installed services, open TCP and UDP ports (used by a service on both the client and server side to make a connection), IP and MAC address of each interface. We will also document administrative credentials, physical site location, and where the server plugs into the network. For virtual servers, we will document the hypervisor administrative credentials and the connected vSwitch connecting the server. Furthermore, we need to document the Network Functions Virtualized (NFV) chain connecting each and every virtual server to a physical Network Interconnect (NIC) on the hypervisor, and then where the hypervisor NIC is plugged into the network.

In the third step, we will identify attributes of each and every network device connecting client hosts to server to any edge device. Here, we want to examine the make and model of the device, the administrative credentials, the configuration of the device, such as how many network ports and of what type (such as 1G or 10G ports) and where those ports are connected in respect to other devices. Thus, we will draw a network map based on discovered elements. Within each device, we will need to document the configuration of the device, and assign the configuration a version number, such as version 1.0. This allows us later to make controlled changes so that we understand what is valid and not valid. This step will begin out direct chain of custody of configurations.

In the fourth step, we will identify what zones exist within the network. This tends to flow along the workflow of the company, such as "Engineering" or "Accounting." We will also document each and every remote site, and treat the WAN as its own zone. Understanding how the network is logically broken up will help us later to identify priorities and severity levels, as well as border areas for security checks.

In the fifth step, we will scan the network for our information and data assets as well as develop a data object sensitivity classification system that is particularly critical to overall network security. This step not only is purely technological, but also involves understanding your specific companies' terms and sensitivities. The process of scanning the network will involve scanning of individual host storage (like your users' local hard drive), as well as CIFS network shares, FTP sites, NFS mounts on server, as well as newer technologies such as virtual cloud storage. Furthermore, we will document internal workflow data that resides on databases as well as critical internal records such as product protocol types, or employee records. Last, we must document how information is created, its flow though the network, and storage habits of the users.