ENGINEERING-PDH.com
ONLINE CONTINUING EDUCATION

CYBER INFRASTRUCTURE PROTECTION - VOL 1 - PART 3 OF 3

Main Category:	Electrical Engineering
Sub Category:	-
Course #:	ELE-146
Course Content:	127 pgs
PDH/CE Hours:	8

OFFICIAL COURSE/EXAM

(SEE INSTRUCTIONS ON NEXT PAGE)

WWW.ENGINEERING-PDH.COM
TOLL FREE (US & CA): 1-833-ENGR-PDH (1-833-364-7734)
SUPPORT@ENGINEERING-PDH.COM

ELE-146 EXAM PREVIEW

- TAKE EXAM! -

Instructions:

- At your convenience and own pace, review the course material below. When ready, click "Take Exam!" above to complete the live graded exam. (Note it may take a few seconds for the link to pull up the exam.) You will be able to re-take the exam as many times as needed to pass.
- Upon a satisfactory completion of the course exam, which is a score of 70% or better, you will be provided with your course completion certificate. Be sure to download and print your certificates to keep for your records.

Exam Preview:

1. Data centers can play different roles for cyber legacy infrastructure. Data center sites can be classified as having one of two main roles: Active sites or Disaster Recovery sites.
    a. True
    b. False
2. Which of the following properties of a well-designed data center allows for variations in the data volume should not affect the DC's quality of service?
    a. Flexibility
    b. Manageability
    c. Scalability
    d. Availability
3. According to the reference material, Data Backup is the act of increasing the number of database servers which are available for clients, mainly for load balancing.
    a. True
    b. False
4. Using Figure 10.6 Vulnerability Breakdown Based on Effect and the surrounding reference material, what percentage of problems lead to a denial of service in either a end-device or a server?
    a. 12 %
    b. 13 %
    c. 20 %
    d. 48 %

5. According to the reference material, A Denial of Service (DoS) attack is defined as an attack that prevents a network or a computer from providing service to the legitimate users.
    a. True
    b. False
6. H.___ is an ITU defined protocol family for VoIP (audio and video) over packet-switched data networks. The various sub protocols are encoded in Abstract Syntax Notation One (ASN.1) format.
    a. 323
    b. 225
    c. 332
    d. 433
7. According to multiple reports released in the beginning of 2008, the number of unique malware files is increasing at an alarming rate. One study said it found almost ___ million unique files, up from approximately 973,000 in the previous year.
    a. 4.5
    b. 5.5
    c. 6.5
    d. 7.5
8. Using Figure 10.9. Vulnerability Breakdown Based on Source, what percentage of problems arise from the implementation issues?
    a. 15 %
    b. 7 %
    c. 91 %
    d. 48 %
9. According to the reference material, device configurations have an average of ___ lines of code per device. A Fortune 500 enterprise that relies on an IP can easily have over 50 million lines of configuration code in its network.
    a. 200
    b. 300
    c. 400
    d. 500
10. NAS are data storage devices that are connected-directly to the network with their own IP addresses, while SANs are storage devices which are connected to each other and connected to a server or a group of servers which act as access points for clients.
    a. True
    b. False

CONTENTS

PREFACE

The Internet, as well as other telecommunication networks and information systems, have become an integrated part of our daily lives, and our dependency upon their underlying infrastructure is ever-increasing. Unfortunately, as our dependency has grown, so have hostile attacks on the cyber infrastructure by network predators. The lack of security as a core element in the initial design of these information systems has made common desktop software, infrastructure services, and information networks increasingly vulnerable to continuous and innovative breakers of security. Worms, viruses, and spam are examples of attacks that cost the global economy billions of dollars in lost productivity. Sophisticated distributed denial of service (DDoS) attacks that use thousands of web robots (bots) on the Internet and telecommunications networks are on the rise. The ramifications of these attacks are clear: the potential for a devastating large-scale network failure, service interruption, or the total unavailability of service.

Yet many security programs are based solely on reactive measures, such as the patching of software or the detection of attacks that have already occurred, instead of proactive measures that prevent attacks in the first place. Most of the network security configurations are performed manually and require experts to monitor, tune security devices, and recover from attacks. On the other hand, attacks are getting more sophisticated and highly automated, which gives the attackers an advantage in this technology race.

A key contribution of this book is that it provides an integrated view and a comprehensive framework

of the various issues relating to cyber infrastructure protection. It covers not only strategy and policy issues, but it also covers social, legal, and technical aspects of cyber security as well.

We strongly recommend this book for policymakers and researchers so that they may stay abreast of the latest research and develop a greater understanding of cyber security issues.

PART III:

TECHNICAL ASPECTS

CHAPTER 8

RESILIENCE OF DATA CENTERS

Yehia H. Khalil*
Adel S. Elmaghraby*

INTRODUCTION

Data centers (DC) are the core of the national cyber infrastructure. With the incredible growth of critical data volumes in financial institutions, government organizations, and global companies, data centers are becoming larger and more distributed posing more challenges for operational continuity in the presence of experienced cyber attackers and occasional natural disasters. The need for resilience assessment emerged due to the gap in existing reliability, availability, and serviceability (RAS) measures. Resilience as an evaluation metric leads to better proactive perspective in system design and management. An illustration of the need for resilience evaluation and a survey of relevant research are presented.

Many organizations now depend on their ability to access their data for their daily operations. Despite the increased power of personal computers and departmental workstations, we notice an increased dependency on centralized data centers due to the needs for data integration, data consistency, and data quality. With the enormous growth of critical data volumes

* This work was partially funded by a grant from the U.S. Department of Treasury through a subcontract from the University of Kentucky. The opinions and conclusion in this paper are the sole responsibility of the authors.

for financial, global, institute, and governmental organizations, data centers have evolved to become the key nerve center of the operations of these organizations. A data center is a facility used for housing a large number of servers/workstations, data storages devices, communications equipment, and monitoring devices as shown in Figure 8.1. The complex architecture of a data center and the variety of data types hosted or processed in a data center complicates their design, planning, and management.[1]

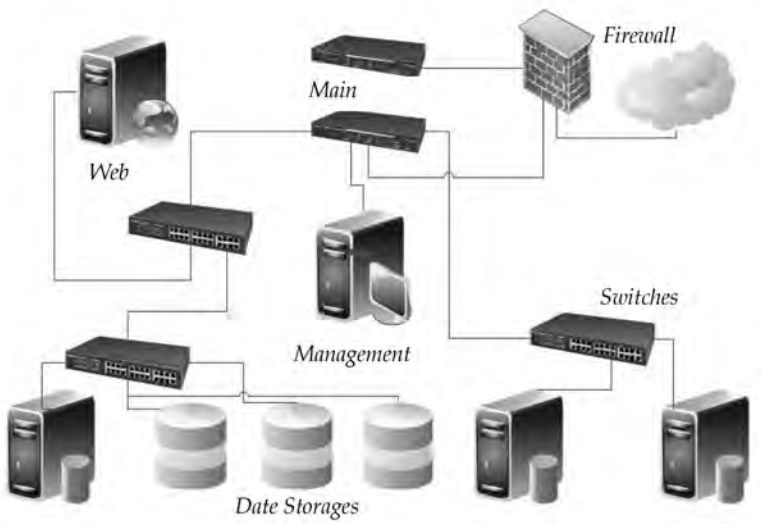


Figure 8.1. A Typical Data Center.

The fundamental purpose of any data center is to furnish application data access requirements; data center design must include operational requirements such as:

- Physically secure and safe location.
- Afford reliable and dependable power supply.
- Healthy environment to run these devices safely.

- Afford communications within the data center and with the outside world.

A well designed data center will have the following properties:

- Flexibility: The ability of a DC to support new applications, services, and hardware substitution without major technology compatibility problems.
- Availability: There is no room for risk with critical applications so a DC should be in a good running condition all the time and maintain a high service level without any unplanned shutdowns related to hardware failures.
- Scalability: Variations in the data volume should not affect the DC's quality of service.
- Security: It maintains different factors; physical, operational, communication network, data storage, and application security.
- Manageability: Simplicity makes it easier for technical support, administration staff, and for troubleshooting errors.[2]

In many scenarios, the data center is used as a standalone facility which is not always the case; data centers can play different roles for cyber legacy infrastructure. Data center sites can be classified as having one of three main roles:

1. Active Site: The main data center which processes all client requests and maintains local data backups.

2. Stand-by Site: This data center is ready to process client requests at any point of time if any request was redirected to it for load balancing. It is connected to the active site thought fiber optics to perform synchronous data replication and it is located within a small distance from the active site.

3. Disaster Recovery Site: It is located geographically far away from the active site for security reasons; it is not ready to process user clients while the active site is up; and it is connected to the active site to perform asynchronous data replication.[3]

In some scenarios, those roles can be combined based on system requirements and the need to implement designer goals and objectives as shown in Figure 8.2.

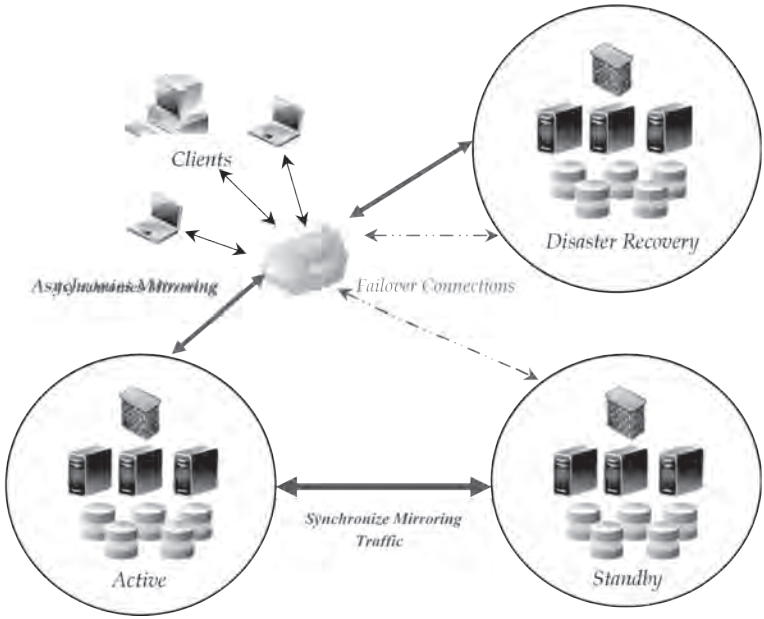


Figure 8.2 . Data Center Roles Summary.

To ensure the operational continuity of critical applications it is mandatory for the data center to provide satisfactory levels of data availability, integrity, and consistency. Yet the growing challenges that data centers face necessitate new methodologies and ap-

proaches to ensure data center operational continuity, and threats like natural and man-made disasters and industrial spying elevate the necessuty for extremely resilient data centers. Elements of rational data centers include computer networks, data storage, security, and data mirroring as shown on Figure 8.3.

Figure 8.3. Data Center Rational Elements.

Cyber system infrastructure evaluation is a significant process toward systems enhancement and management. Conventional computer system evaluators intend to examine levels of RAS for their systems where:

- A reliable system does not deliver results that include uncorrected corrupted data, and it works to correct the corruption if applicable or shuts the system down.
- Availability is the uptime of device operating as the percentage of total time.
- Serviceability measures the ease to maintain, diagnose, and repair the system.

With new and emerging technologies, system complexity, data volumes, and new threats confirm the need for novel methodologies and approaches to assess of the resilience of data centers.

Resilience is the ability of a system to resist illegitimate activity and its ability to effect a speedy recovery as shown in Figure 8.4.[4] The main aspect is to show how the system will be affected by the variation of the operational environment circumstances. However, it is used quite differently in different fields, for example, computer network resilience is the ability of the network to provide and maintain an acceptable level of service under different fault or an abnormal conditions caused by cyber threats or any other threats. While in business, resilience is the ability of a company, resource, or structure to sustain the impact of a business interruption, recover, and resume its operations to continue to provide minimum services.

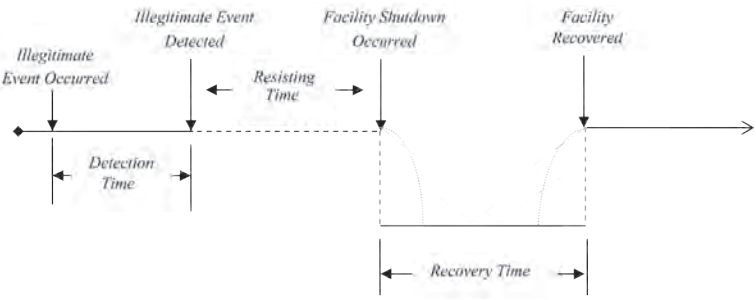


Figure 8.4. DC Facility Failure Process Summary.

The current data center metrics cover many of the concerns of data center designers and mangers, however there is still another set of concerns that lacks answers. All of the current metrics evaluate systems while they are operating or after a failure has oc-

curred. A proactive metric is needed to evaluate a system during all of its stages: launched attacks, resisting/adaptation to attacks, failure and recovery time, and patterns.

STATE OF THE ART

Storage Research and Technologies.

Data storage is an integral part of the architecture of a data centers, over the years several storage solutions have been developed to satisfy applications requirements and demands.[5] Storage Area Networks (SAN) and Network Area Storage (NAS) are dominating data center storage alternatives.

NAS are data storage devices that are connected-directly to the network with their own IP addresses, while SANs are storage devices which are connected to each other and connected to a server or a group of servers which act as access points for clients.[6] Table 8.1 presents preliminary guidelines for a storage solution selecting process.

Criteria	SAN	NAS
Cost	Expensive	Inexpensive
Setup	Complicated	Straightforward
Management	Easy	Complicated for large environment
Environment size	Better for large	Better for small
Disk system compatibility	Any	Device orientated
Impact on network	None	Can swamp a network

Table 8.1. SAN vs. NAS Summary.

SAN setup costs are getting cheaper and with less complicated management, so the future of SAN is more promising for data centers which make it the focus of this work. Rationally, SAN solutions have two components: *storage servers and storage clients*. The physical elements of SANs are as shown in Figure 8.5:

1. *Disks* can be connected as point-to-point without an interconnection device or they can be a part of server-storage model. SANs are independent from disk types; disks, tapes, RAIDs, and file servers can be used.

2. *Servers* are fundamental elements of a SAN, which can be a mix of platforms and OS.

3. *Communications* are implemented by a fiber channel, where data loss rate is zero, and there is a high throughput rate.

Figure 8.5. Elements of Storage Area Networks.

Regardless of which type of storage technology is used within data center, storage devices are required to: support data *Access; Protect; Store; Move; and Recover* with minimum cost (management and setup). Traditionally researchers and engineers utilized *Latency* as a performance metric, which naturally characterizes hard drive access time. Input/Output (I/O) issues are a critical aspect of any storage solution, the gap between the server processing rate and the I/O rate is large. SAN producers aim to overcome this limitation by improving Cache memory size and Caching algorithms. Yet, on the other hand, latency did not show how storage solutions protect & recover data, hence resilience metrics are required.

It is highly recommended designating between two categories of the data hosted in a data center:

1. *In use data:* which are accessed, modified and updated.
2. *In rest data:* which are not used at the moment and only stored to be used later or for recovery purposes.

Normally, "Data In Rest" is easier to protect and recover while "Data In Use" requires more effort. Data storage solution resilience evaluation results in the following concerns:

1. Availability of alternative routing paths within the fabric cloud.
2. Routing protocols' capability to utilize an alternative path with minimum cost (converging time and routing table size).
3. Optimum number of local disk images, backups to ensure fast recovery.
4. Data backup/ mirroring process frequency.
5. Mirroring approach impact on response time.
6. Protection techniques strength and overhead.

A resilient storage solution will not only provide the optimum answer for the concerns highlighted above, but will also tune them up jointly, to achieve the maximum resilience level.

Data Mirroring Techniques and Methodologies.

Data mirroring, data replication, and data backup are popular terms used in the context of data availability, consistency, and recovery. It is vital to differentiate their usage, limitation, and challenges to utilize them for a resilient data center.[7] By definition, *Data Backup* is the process of copying data (files/databases) into other data storage to be retrieved when needed in case of device failure. It is considered a regular process of system management and usually done overnight, which means a full working day's data may be lost in case of a device failure. For critical applications, this amount of lost data is unacceptable.[8]

Data Replication is the act of increasing the number of database servers which are available for clients, mainly for load balancing. Data replication can be done within or off the facility. For speedy recovery and critical data application, *Data Mirroring* is mandatory. Data mirroring is the copying of data from one location to a storage device or different location in real time. It is always a good idea to have the mirroring site a safe distance from the main site.[9]

For resilient data centers, in addition to data replication, data mirroring is mandatory. Data mirroring can be implemented as synchronous or asynchronous. In the case of synchronous mirroring, each transaction is sent to the mirror site and the clients do not get a response until the main site gets acknowledgment from the mirror site as shown in Figure 8.6. This approach

affects the system performance and increases service response time.

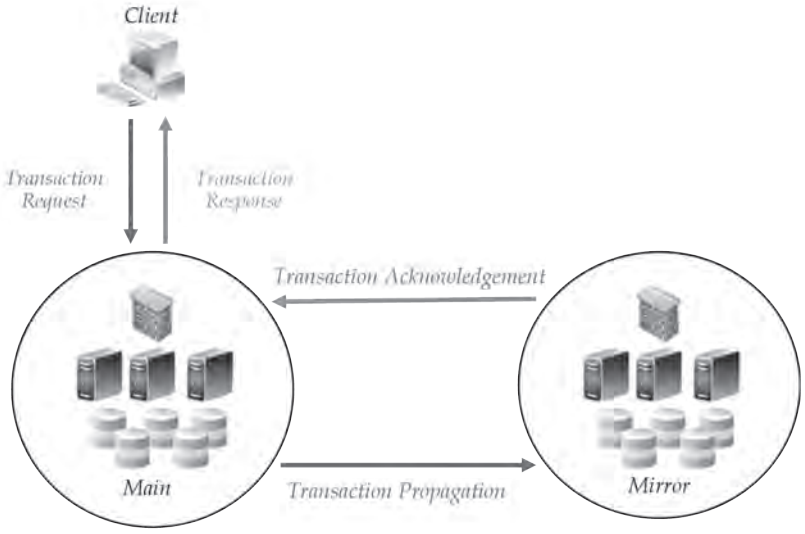


Figure 8.6. Synchronous Data Mirroring Process.

Also, data mirroring can be implemented as asynchronous where the main site receives the client's request, processes it, responds to the client, and then sends updates to the mirror site as shown in Figure 8.7. In this case, the mirror site will be a few transactions behind; but the system performance will not be affected.

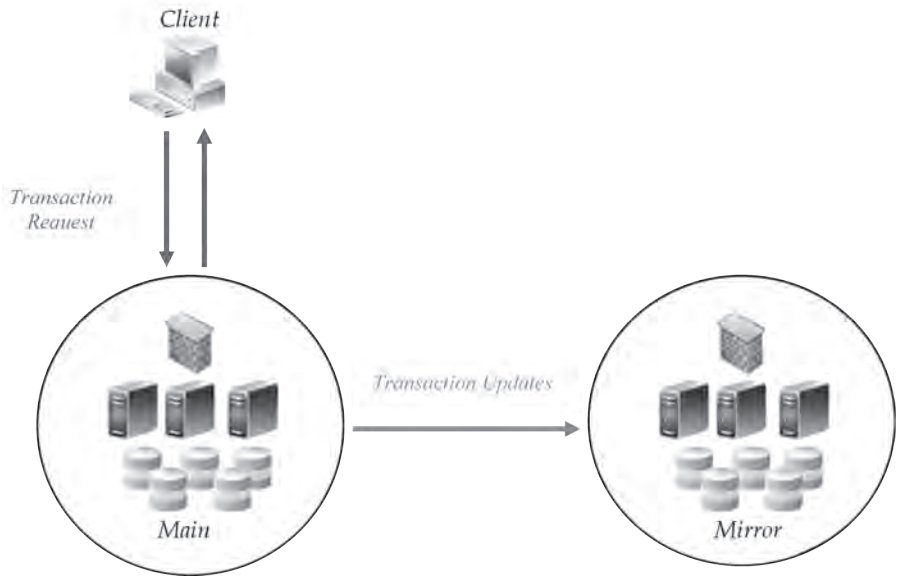


Figure 8.7. Asynchronous Data Mirroring Process.

Many database mirroring technologies are available on the market which require detailed investigation for the following aspects to simplify selection processes:

1. Supported Platforms: Multiple platforms provide more flexibility for data centers.

2. Change Check Capability: Current tools focus on only the net data change after the last mirroring process occurrence.

3. Computability Issues: Most of the current tools work as "plug-in" with no change required for database scheme and support several network topologies (server-server, hub-and-spokes).

4. Public Networks: New challenges for mirroring tools are raised when data are transmitted over public networks. These include: data security, TCP/IP vulnerability, and firewalls.

5. Scalability: Adding/removing sites is always needed regardless of how easy the reconfiguration process is.

For resilient data center systems, remote sites are mandatory which infuses the need for powerful mirroring tools over public networks/Internet where a "hand-shaking" process requires more effort. Also sequential data block transmission is not appropriate because of the public network/Internet nature. IBM Global Mirroring employs flash copy technology which permits data blocks to be partitioned into smaller portions by sending them to the mirror site, reassembling it, and writing it to the database.

Mirroring time is the time required to mirror data to the remote site while pause time is where the mirroring tool is inactive. For many mirroring tools, those parameters can be controlled either by direct or indirect ways. Figure 8.8 illustrates two scenarios that show how critical the tuning of those parameters can be in combination with the detection time of malicious activities for system resilience.

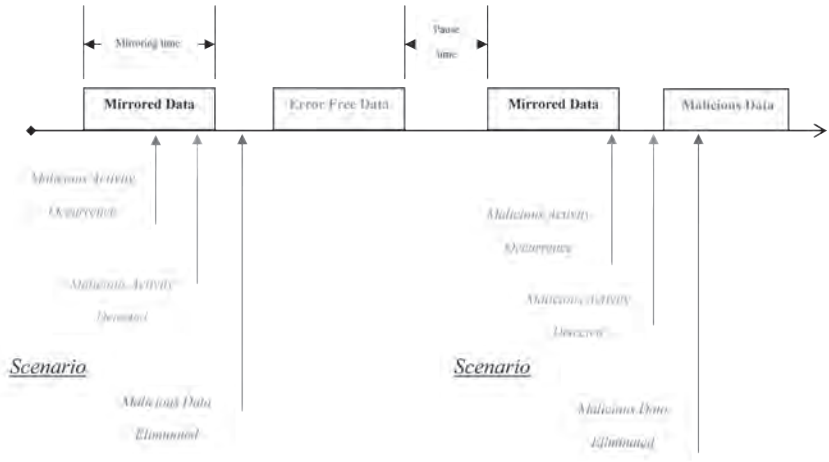


Figure 8.8. Data Mirroring Parameters and Attack Scenarios.

In scenario A, data mirroring parameters worked together with malicious activities detection time to ensure that the data sent to the mirror site were error free, while in scenario B the parameters were not correctly tuned, which resulted in sending malicious data to the mirror site. Mirroring corrupted data to the recovery site ruins the objectives of the recovery site and system resilience.

Network Connectivity Alternatives.

Network connectivity represents a significant portion of data center architecture, for Interconnection, Data Storage, Mirroring, and Public Access as shown in Figure 8.9. Also, computer network subelements (topologies, links, connecting devices, routing protocols and load balancing) are very critical aspects of computer network performance and resilience level.[10]

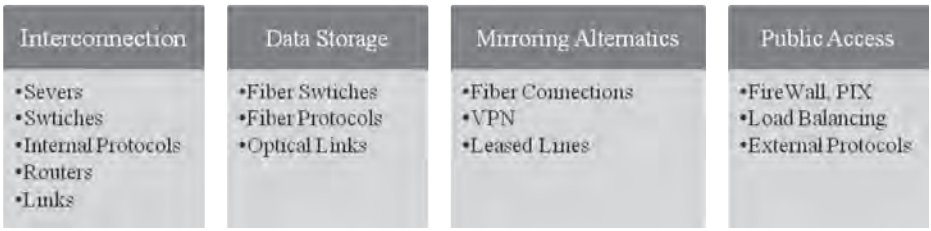


Figure 8.9. Data Center Network Roles Summary.

Resilience, *Redundancy,* and *Fault Tolerance* are widely used terms within computer network assessment and design context. For decades network designers and analysts used redundancy to improve network availability and reliability. It is essential to define each term. *Redundancy* is the process of installing extra equipment to overcome any node failure.

It requires having a plan and tools to direct traffic through the replacement node, thus redundancy cannot enhance network resilience by itself. The main idea of *Fault Tolerance* is to recognize how a node or device can fail and therefore take the necessary steps to prevent the failure. The definition of *Resilience* is the ability of a system to maintain any disturbance with minimum change on performance efficiency, and to effect a speedy recovery from any disturbance.

For critical applications and legacy systems, connectivity failure is an extremely hazardous situation because it revokes system integrity and operational continuity. In addition, the nature of connectivity failures is different than other computer system failures in the following aspects:

1. *Detection*: In some cases it is a complicated process. For instance, chronic failure detection is harder than sudden (crash) failure detection.

2. *Cascading Nature:* A simple failure can affect a large number of services or clients. In addition, this simple failure can overload other parts of the network, causing the system to crash resulting in more harm.

3. *Origin*: The same failure can be originated by many factors, for example, a node failure can be caused by power outage, software failure, or malicious activities.

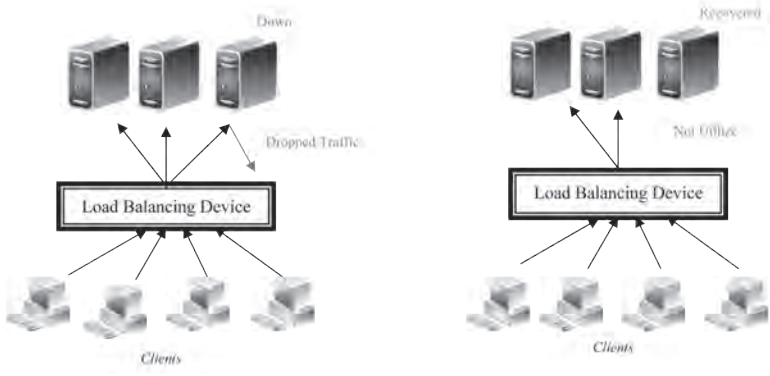
Network resilience assessment comprises two main aspects:

1. Alternatives: It is very critical to have alternatives for network elements which are not limited only to redundant equipment but also include alternative traffic routing paths. The alternative paths must be reserved only for major failures and ensure *approximately* the same cost of original route in terms of laten-

cy and response time without causing other network parts failures.

2. Recovery Tools: Redundancy is mandatory for resilience, in addition to tools to employ those devices in timely manner; routing protocols (RP) and load balancing algorithms (LBA) are fundamental tools to ensure network resilience. The ability of RP to utilize alternative routing paths with minimum converging time is essential for a resilient network. The ability of LBA to detect failed/recovered servers in a minimum amount of time is a critical issue for network resilience.

Consider the following scenario; a system that is using LBA with less than optimal parameter tuning as shown in Figure 8.10. In the first case traffic is sent to the server while it is down, in the second case no traffic is sent to server after recovery. In both cases it offers poor resilience level for server failure and recovery.[11]



Case 1: Down Server undetected. Case 2: Recovered Server undetected.

Figure 8.10. Load Balancing with a Poor Parameters Tuning Scenario.

One of the vital parameter routing protocols challenged by multiple failures of network parts, is the route cost. Consider traffic sent from point A to point B; assuming that main route R1=(x1, x2, x7) and the alternatives routes are R2=(x4, x5, x6) and R3=(x4, x3, x7) as shown in Figure 8.11. It is clear that x4 is common for both alternative routes which is not conventional particularly in the case of an x4 failure. Also alternative routing costs might lead to an overload of certain parts of the network causing even more failures.

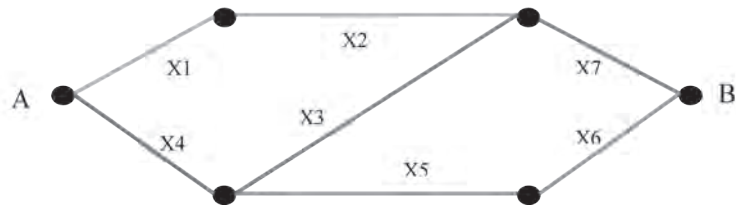


Figure 8.11. Routing Protocols Concerns Summary.

Security Challenges and Opportunities.

The fact that the data center is the core of any legacy system and it hosts large critical data volumes make it a target for all type of attacks, physical or cyber. Surveillance cam, high-tech doors, and other technologies improve the data center's physical security. However, on the other hand, data center cyber security is a much more challenging process.[12]

Potential network *Vulnerabilities, Threats, and Attacks* must be identified to minimize security concerns. System *Vulnerabilities* refer to weaknesses in the system that can be attacked, while *Threats* are the potential to cause damage to data center resources. *Attacks* are the

actual use of system *Vulnerability* to put *Threats* into action. System hacking is a continuous process where hackers continue to discover system vulnerabilities to develop attacks as depicted in Figure 8.12.

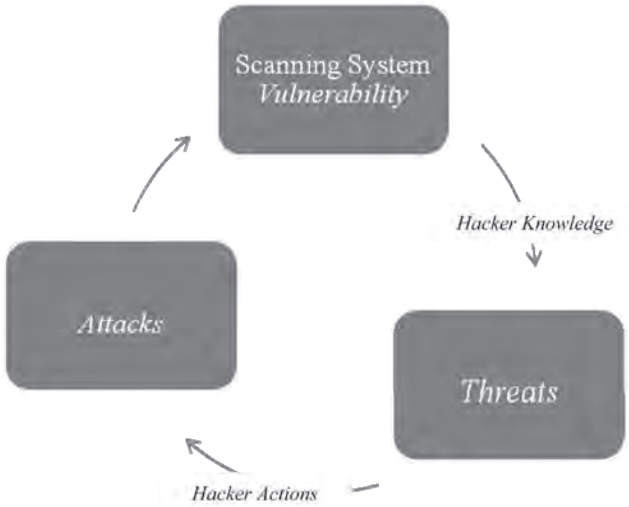


Figure 8.12. Developing Attacks Process.

Enumerating all possible data center vulnerabilities, threats, and attacks in an exact list is not feasible, yet they can be categorized as Table 8.2 shows.

Vulnerabilities	Threats	Attacks
Designing	Intrusion	Denial of Service (DoS) and Distributed DoS (DDoS)
Technologies	Spam	Un-authorized Access
Applications	Worm	Information Tampering
Database	Virus	Cross-site Scripting
Networks	Malware	IP Spoofing
Monitoring tools	Spyware	Insider Malicious Activities

Table 8.2. Vulnerabilities, Threats, and Attacks Categories Summary.

Even as the hacker is working hard to elude data center security, data center designers, venders, and security teams are working just as hard to ensure data center safety and security. Their efforts have produced many technologies such as firewalls, intrusion detection and prevention tools, DoS and DDoS detection and mitigation, access lists, and access restriction.

Data center security has three layers: (1) networks, (2) applications, and (3) databases. Figure 8.13 demonstrates the security mechanisms that are currently available.

IDS: Intrusion Detection Systems
IPS: Intrusion Prevention Systems
NAC: Network Admission Control
ACL: Access Common List.

Figure 8.13. Security Layers Summary.

Security evaluation is done for different purposes:

1. Products, organization, application accreditation.

2. For the development and enhancement of security policies, methodologies, and technologies.

Researchers and system developers focus on the second perspective: Legacy system security assessment is a very complex process for many reasons:

- *Data Characteristics*: Each data type is targeted by certain hackers and attacks, as in the case of financial, military, and industrial data.
- *Data Status*: Data that is in an *Operation* mode is harder to protect, while data in a *Rest* mode requires less effort.
- *System Design*: Used for utilities, manufacturing companies. These systems will have two networks: business and control. This type of design increases system vulnerabilities and requires special arrangements to ensure network isolation.

For a resilient data center, security technologies and methodologies are expected to guarantee system functionality, information assurance, events management, and correlations. Consequently, security policies must ensure a speedy detection process and the ability to utilize system resources to mitigate attack effects.

CONCLUSION

The assessment process of cyber infrastructure security requires a number of metrics including performance, availability, and reliability. The rapid growth of data volumes, increased complexity of cyber infrastructure, and the heightened levels of threat highlight the gap in existing evaluation metrics. Increased awareness for a proactive approach addressing resilience in various systems including data centers demands a new evaluation approach. Resilience measurement of alternative data center designs assesses their ability to face man-made and natural disasters. Data center subsystems must be considered for a comprehensive resilience evaluation in addition to recovery plans and policies. The proposed resilience metric is proactive and assesses system behavior before, during, and after an attack.

ENDNOTES - CHAPTER 8

1. Mauricio Arregoces and Maurizio Portolani, *Data Center Fundamentals,* Cisco Press, 2004, available from *www.cisco.com/web/about/ac123/ac220/about_cisco_cisco_press_book_series.html*.

2. *Ibid*.

3. Kehia H. Khalil, Anup Kumar, and Adel Elmaghraby, "Design Considerations for Resilient Distributed Data Centers," ISCA 20th International Conference on Parallel and Distributed Computing Systems (PDCS), 2007, pp. 51-55.

4. The definition of resilience is available from *www.merriam-webster.com/*.

5. Richard L. Villars, "IBM Total Storage Software: Building Storage Solutions in Alignment with Current and Future Business Requirement," White paper sponsored by IBM, 2004.

6. Gary Orenstein, *IP Storage Networking: Straight to the Core,* Addison-Wesley, 2003, available from *www.pearsoned.co.uk/imprints/addison-wesley/*.

7. Jim´Enez-Peris, R. Pati, M. ˜No-Mart´Inez, G. Alonso, and B. Kemme, *How to Select a Replication Protocol According to Scalability, Availability, and Communication Overhead,* The International Symposium on Reliable Distributed Systems (SRDS), New Orleans, LA: IEEE Computer Society Press, 2001, pp. 24–33.

8. Wang Changxu and Xu Rongsheng, "Analysis and Research of Data Backup System in Corporation," *Computer Applications and Software,* Vol. 25, No. 10, 2008, pp. 121–123.

9. M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso, "Understanding Replication in Databases and Distributed Systems," the 20th International Conference on Distributed Computing Systems, 2000, pp. 464-486.

10. Alberto Leon-Garcia, and Indra Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures,* New York: McGraw-Hill Professional, 2004.

11. Yehia H. Khalil and Adel Elmaghraby, "Evaluating Server Load Balancing Algorithms For Data Center's Resilience Enhancement," New Orleans, LA: ISCA 21st International Conference on Parallel and Distributed Computing and Communication Systems, September 2008, pp. 111-116.

12. Merrill Warkentin and Rayford Vaughn, "Enterprise Information Systems Assurance and System Security: Managerial and Technical Issues," Idea Group Pub., 2006, available from *www.igi-global.com/.*

13. Shoukat Ali, Anthony A. Maciejewski, Howard Jay Siegel, and Jong-Kook Kim, "Measuring the Robustness of a Resource Allocation," *Transactions on Parallel And Distributed Systems*, 2004, Vol. 15, No. 7, New York: IEEE, pp. 630-641.

14. R. Jain, *The Art of Computer Systems Performance Analysis,* New York: John Wiley & Sons, 1991.

15. Denis Trček, *Managing Information Systems Security and Privacy*, Basel, Switzerland: Birkhäuser, 2006.

16. M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "A Large-scale and Decentralized Application-level Multicast Infrastructure," *Journal on Selected Areas in Communication* (JSAC), 2002, New York: IEEE, pp. 20-27.

17. Albert Greenberg, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta, "Towards a Next Generation Data Center Architecture: Scalability and Commoditization," ACM workshop on programmable routers for extensible services of tomorrow, Seattle, WA, 2008, pp. 57-62.

CHAPTER 9

DEVELOPING HIGH FIDELITY SENSORS FOR INTRUSION ACTIVITY ON ENTERPRISE NETWORKS

Edward Wagner and Anup K. Ghosh

INTRODUCTION

Future success in cyber will require flexible security, which can respond to the dynamic nature of current and future threats. Much of our current defenses are based upon fixed defenses that attempt to protect internal assets against external threats. Appliances like firewalls and proxies positioned at network segment perimeters similar to the Maginot Line attempt to prevent outsiders from breaking in. There are other mechanisms such as Public Key Infrastructure and antivirus software, which also provide security. This added layer is referred to as a "Defense in Depth" methodology. However, in each component of our security architecture vulnerabilities are revealed over time. These defenses lack any agility. Our defenses must become agile and responsive.

In large-scale enterprise network defense, intrusions are detected by monitoring network flows from untrusted sources. Primarily, network intrusion detection systems examine traffic at Internet gateways, and then again at individual enterprise units or at enclave routers. Intrusions detected at lower organizational structures are detected then reported to higher reporting entities. The current approach to detecting intrusions suffers from two main problems: (1) intrusion

sensors are placed in locations that do not allow high fidelity examination of intrusion behavior; and (2) intrusions are detected by comparing network traffic to known malicious intrusion patterns. In this chapter, we propose a supplemental method to correct for these two deficiencies. We propose high fidelity sensors in the form of virtualized applications on each user's machine to complement network-based sensors. In addition, we equip each user such that even as they are reporting intrusions, their machine and data is protected from the intrusions they are reporting. Our approach will protect users from broad classes of malicious code attacks, while being able to detect and report both known and unknown attack code.

One of the core strengths of our approach is that most attacks are realized at the endpoint. Attack code is often embedded and obfuscated in network traffic that often flies by network sensors unnoticed. When reaching a vulnerable host, the code is executed. Our contention is that the endpoint (host) is the best place to detect most attack codes because it is at this point that the behavior of the attack codes can be observed. Of course, once the attack code runs, it poses a high potential risk to the host and network, so we virtualize networked applications to protect the host from the attack code.

The user operating environment can remain on the existing infrastructure, but operate in a virtual workspace. Bringing virtual computing to the host will allow compartmentalization of the environment, wherein untrusted applications or applications that run untrusted content are partitioned from the trusted host itself. If the untrusted application environment is compromised, the underlying host will remain uncompromised. We also leverage sensor technology in

the virtualized environment to detect illicit changes and then report these to a database. The environment is then "refreshed" removing any persistent presence of a threat. The environment can be configured and updated frequently, then copied and distributed en mass. Since the provisioning of the virtual environment is done centrally, changes to the environment can be easily identified and captured.

This new environment becomes agile and dynamic. It regains the initiative that the threat has taken and retained for over a decade. Malware can be stamped and collected as it appears. Malicious or compromised websites can be identified and blocked if desired, or monitored for intelligence purposes.

CURRENT ISSUES

According to the Center for Strategic and International Studies report on Cybersecurity prepared for President Barack Obama, potential adversaries have compromised and penetrated computer networks in the United States. These perpetrators have accessed and retrieved large quantities of information. In 2007, the compromises included the unclassified e-mail account of the Secretary of Defense and the exfiltration of terabytes of data from the Department of State. The report says that the loss of government data and intellectual property threatens our economic and national security.[1]

Given this threat environment, there are significant efforts to monitor networks. The current array of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are dependant on the continual development of signatures to find threats operating within the network. IDS are devices that monitor net-

work segments with a set of signatures to match nefarious activity. An alert is created when a match is found. Analysts monitoring these systems must make a subjective decision whether or not to investigate the alert. Even if the information is correlated with other security devices such as firewall alerts in an Enterprise Security Management tool, the assessment of risk is imprecise. The IPS is different because it will drop the packets of the related session when the signatures match. The dropping of packets is the equivalent of disrupting the attack. This prevents the attack from being successful. While there has been continuous innovation to improve the ability to identify threat activity, this capability is limited by the signature's ability to know the method of attack.

The use of antivirus suffers from the same inherent limitations of a signature-based system. The volume of malware is frequently described as an antivirus problem but, in fact it affects the entire system. According to multiple reports released in the beginning of 2008, the number of unique malware files is increasing at an alarming rate. One study said it found almost 5.5 million unique files, up from approximately 973,000 in the previous year.[2] The development of signatures cannot keep pace in a timely manner, neither can the ability to distribute, store, and analyze files on client hosts from these large databases. Thus, signature-based antivirus suffers from the temporal dynamics of rate of change of malware, the broad proliferation of malware, and the scalability of the distribution and storage of antivirus signatures. Attacks will frequently elicit the assistance of users through an e-mail directing them to a website with malware. At other times, legitimate websites are unknowingly compromised. One study of 145,000 commonly visited Chinese websites found 2,149, or 1.49 percent, had malicious content."[3]

The development of signatures requires the investment of resources to analyze attacks to understand the tactics, techniques, and procedures and then develop corresponding signatures. The cycle typically begins with actors identifying vulnerabilities, it next moves to the development of exploit code where defenders are continually seeking the formulation of mitigation strategies to prevent attacks.

There are huge resources poured into the discovery of vulnerabilities. Those who seek information on vulnerabilities are both network attackers and defenders alike. Sutton and Nagle describe the emerging economic market of identifying vulnerabilities in their paper to the Workshop on Economics of Information Security, 2006 titled: "iDefense gains revenue by directly reselling the information, while TippingPoint profits by offering exclusive protection against the vulnerabilities they purchase via their intrusion detection system (IDS) product."[4] Frequently, observers focus on the price paid as a result of cyber attacks, but few recognize the transactions occurring to develop signatures and other network defense measures versus the effort to exploit them.

Large organizations spend considerable resources to maintain their IDS/IPS infrastructure and the corresponding signature base. Smaller organizations frequently outsource some of the effort through signature subscription services. In order to support the development of signatures, there is a heavy dependence on analytical work to find anomalies, collect malware, reverse engineer them, and finally develop signatures.

Companies that provide this service do so in two ways. Some companies provide a fee for service. They rely on their own collection infrastructure to collect malware traversing the Internet. They may use a network of "honeypots" to collect the malware. Once

collected they analyze the malware and develop a signature. These signatures and the associated threat warnings are then provided to their customers.

Other companies provide analytical support directly to organizations and rely on the collection infrastructure of the supported organization. Many organizations object to completely outsourcing their security services due to information disclosure concerns. The Department of Defense follows this example. The collection of information and resulting analysis is complex and laborious. Colonel Barry Hensley, Director of the Army Global Network Operations Security Center, described the growing demand for forensic analysis at the Army LandWarNet conference in 2008. He said, "People don't realize the forensics handling process involved with identifying malicious code. . . . It can take weeks or months."[5]

Consumers of signature support and services finds it difficult to measure the effectiveness of their purchase. There are many key questions for example: How many attacks were never detected because a signature was never developed? Were any of the signatures ignored by a poorly trained IDS analyst?

Victor Oppleman, Oliver Friedrichs, and Brett Watson further describe the breadth and width of the problem with IDS/IPS in *Extreme Exploits: Advanced Defenses Against Hardcore Hacks.* They identify three significant reasons for the problem: (1) "The granularity of the signature syntax," (2) "Whether the signature detects a specific exploitation of a vulnerability or the core vulnerability itself," and (3) "The author of the signature and the protocol knowledge he possesses."[6]

Some signatures are more effective than others. Some signatures will alert to real threats as well as other traffic, which is not actually an attack. Frequently this

is referred to as a false positive rate. A more granular signature can frequently address this problem. Some signatures are written based on known exploit code, but there may be other exploit codes that attack the same vulnerability, but do not alert the same signature. Finally, the effectiveness of a signature can be a reflection of the skill and knowledge of its author.

The problems with IDS/IPS do not simply stem from the development and deployment of effective signatures. Potential attackers are constantly looking for ways to avoid detection. In "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," Ptacek and Newsham provide three examples of how threats will attempt to elude IDS at the network (IP) and transport (TCP) layer. The three examples are "IP Checksum, IP TTL, and IP Fragmentation."[7]

In the IP Checksum example, the method data is verified when it travels across the Internet and can be manipulated to confuse an IDS/IPS. If the IDS/IPS does not conduct a checksum validation, it will accept data in an invalid packet, which would normally be dropped. This may result in fake data being used to throw off packet inspection.

The second method is when an IDS/IPS accepts a packet with an invalid Time-To-Live (TTL) value in the IP header. Normally the endpoint would drop the packets with an invalid value; however, the IDS/IPS placed at various points in the architecture may accept packets with an invalid value. The result is the complication of the data inspection process.

The third example of obfuscation is IP Fragmentation. Threats can cause an IDS/IPS to accept inserted or crafted fragmentation packets for inspection that would normally be discarded by the endpoint. In each of these examples, the threat is adding packets for

inspection that are used to confuse the inspection of other packets used in an attack.

These examples are just three ways that threats can avoid detection by signature-based IDS/IPSs. There are many more, and the number of attack techniques are only limited by the imagination of the attacker. The limitations of signature-based IDS/IPSs described thus far do not consider the evolutionary nature of the attack themselves. The shift from targeting hosts with exploit codes to targeting users with phishing e-mails highlights the difficulty in detecting attacks.[8] The attacker may target individual users or large groups of users with malware attached to an e-mail message. The malware can compromise the host and initiate a communication request to an external host controlled by the attacker. This reverses the attack sequence that IDS/IPSs look for when an external host attacks an internal host.

If an attacker establishes an encrypted connection between his jump off point and the compromised friendly host, the traffic is never assessed. IDS and IPSs are not able to decrypt such traffic, no signature will cause an alert and the analyst is never able to assess the connection.

THE VIRTUAL ENVIRONMENT

Using the innovation of the Internet Cleanroom, developed previously at GMU,[9] the user can operate in a virtualized environment. This provides the first level of protection by isolating the user from the underlying host. Tools to monitor and collect information about threats operate in the separate host operating system (OS), which provides integrity to the collection of threat information. In signature based monitoring efforts, the volume of data is enormous and unmanage-

able. Since the virtual environment presents a known good, changes from that state can easily be identified and logged. This reduces the volume of monitoring data.

Figure 9.1 displays the architecture of the Internet Cleanroom. It shows the separation of OSs, which is the basis for the reliability of collected data. It also shows the ability to compartmentalize the user's environment.

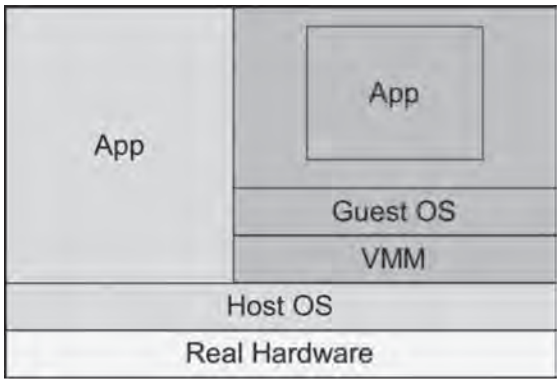


Figure 9.1. Internet Cleanroom Architecture.[10]

The collection architecture that is possible in this environment is displayed in Figure 9.2. Segmentation of the collection mechanism and user environments is achieved through the use of virtualization. Another benefit to collecting compromised URLs at the host is the easy identification of the host involved in the incident. Currently, the collection of data occurs at the network perimeter. The Domain Name Service (DNS) architecture begins below the collection point and the volume of name resolution prevents logging, therefore many hosts visiting known compromised websites cannot be identified. Even if the URL is identified and blocked a host possessing malware may operate on the network indefinitely without remediation.

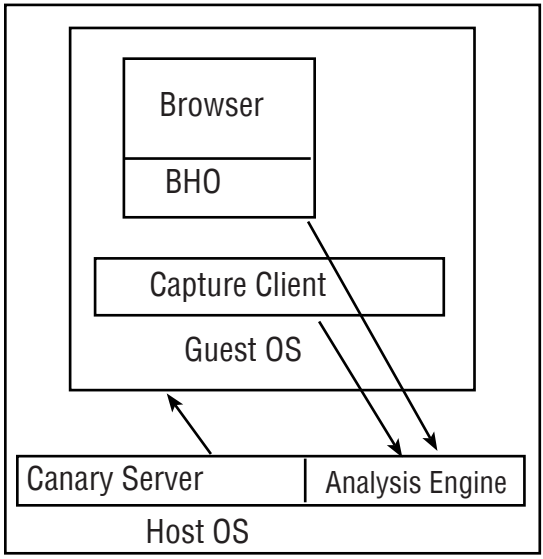


Figure 9.2. Collection Architecture.

The Internet Cleanroom is best deployed to coincide with an existing security architecture as shown in Figure 9.3. To maximize protection, monitoring at the enclave access points should continue. As noted before, the use of traditional IDS/IPS tools remain limited in their ability to detect new and sophisticated threats. Alternatives like extensive review of router logs and netflow can be very helpful. The integration of the Internet Cleanroom into the Security Information Manager architecture allows the correlation of host data from the Internet Cleanroom with network alerts from IDS/IPS. The combination of these two technologies brings more accurate alerts to the SIM analyst. In the past, information security professionals have desired to have access to full packet capture and host logs in near real time. However, the data storage requirements outweighed the usefulness of the

data. Additionally, it overwhelms the analyst with too much data to review. The Internet Cleanroom's ability to gather specific information about threats as it is integrated into a SIM enables greater responsiveness for network defense.

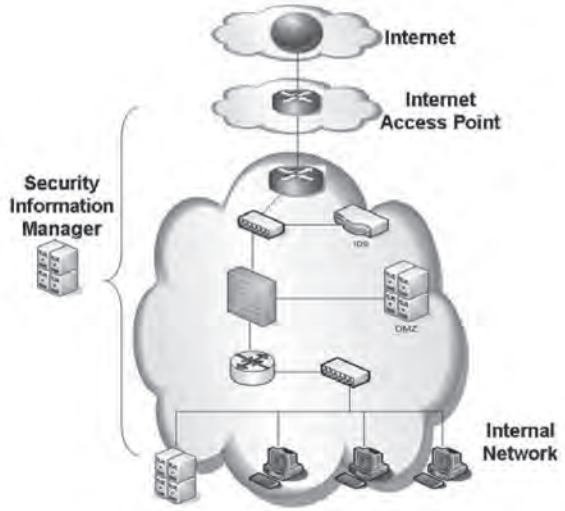


Figure 9.3. Security Architecture.

Making Granular Data Capture Meaningful.

The Internet Cleanroom addresses the capture problem by logging change data. Since the host is provisioned in a VM Client, changes can be easily logged and sent to a SIM via a syslog. Though the number of host remains large, smart data capture makes this type of collection feasible.

Figure 9.4 shows the analyst's view of summary information for hosts including infections that were downloaded and infected websites. The infected websites information provides actionable information in a more timely manner. This information can be provided immediately to any program to block access to those websites.

Figure 9.4. Analyst's Report View.

While the bad URLs remain of interest, the MD5 Hash of any malware downloaded by the host is of great interest to the defenders of the network. To gather this information typically requires a forensic collection on an individual host. This is a manual and time intensive process. However, much of this collection can now be automated in the Internet Cleanroom.

As noted in Figure 9.5, the Internet Cleanroom is able to automate the collection of MD5 Hash of malware and that information is immediately available to the analyst. Instead of the lengthy analysis of forensic media to collect this information, it can be obtained as it happens. If shared with scanning tools, which search for malware by MD5 Hash, network defenders can be more responsive in their ability to identify infected hosts.

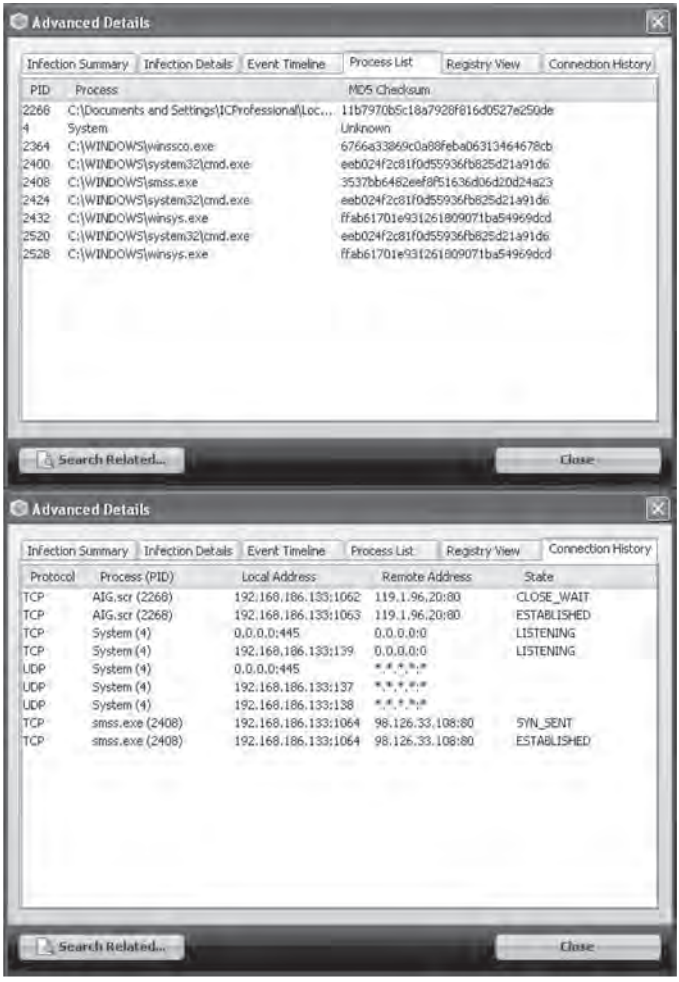


Figure 9.5. MD5 Hash of Malware.

The discovery of malware in near real time can assist in the recognition of new threat trends. Adjustments to perimeter defenses can be made before the loss of the initiative. This type of dynamic defense changes the static intransient defense that has been unable to respond in time to developing threats.

CONCLUSIONS

Current defenses largely rely on signature-based mechanisms in the network or on the host to detect attacks. These techniques have become largely ineffective as the proliferation of malicious software shows. The primary reason for their ineffectiveness is because malware changes its signatures faster than the mechanisms have been developed to capture malware, and then create and distribute those signatures. In addition, we argue that network-based sensors are not adequate for detecting threats against networks. To address these deficiencies, we propose that the enterprise computer network defense architecture should include a virtualized application solution that: (a) protects users from unknown future infections (signature-free defenses), and (b) provides detecting and reporting of unauthorized system changes to a collection database.

The most obvious example of our proposed approach is a virtualized browser that users employ just as they use their native browser. The proposed virtualized browser, Internet Cleanroom, protects the user from malicious web content, while also monitoring the virtual OS for any unauthorized changes that may occur as a result of browsing. The virtualized architecture effectively partitions untrusted applications and content from the underlying operating system and other applications. Any unauthorized changes are noted, then the virtual OS is discarded, and a pristine environment restored, all without any virtualization expertise required. The proposed approach provides high-fidelity detection of malicious code threats that can be later analyzed in forensic detail, while also protecting the user from currently unknown threats.

ENDNOTES - CHAPTER 9

1. Co-Chairs, Representative James R. Langevin, Representative Michael T. McCaul, Scott Charney, Lieutenant General (USAF, Ret.) Harry Raduege, and Project Director James A. Lewis, "Securing Cyberspace for the 44th Presidency," Washington, DC: Center for Strategic and International Studies, December 2008.

2. T. Wilson, "Malware Quietly Reaching Epidemic Levels: New Reports Say Malware Increased by a Factor of Five to 10 in 2007," *Dark Reading*, January 16, 2008, available from *www.darkreading.com/security/vulnerabilities/showArticle.jhtml?articleID=208803810*.

3. Jianwei Zhuge, Thorsten Holz, Chengyu Song, Jinpeng Guo, Xinhui Han, and Wei Zou, "Studying Malicious Websites and the Underground Economy on the Chinese Web," Workshop on the Economics of Information Security, WEIS, 2008, available from *weis2008.econinfosec.org/papers/Holz.pdf.*

4. Michael Sutton and Frank Nagle, "Emerging Economic Models for Vulnerability Research," Workshop on the Economics of Information Security, WEIS, 2006, available from *weis2006.econinfosec.org/docs/17.pdf*.

5. Wyatt Kash, "Army cyber ops faces forensic backlog," *Government Computer News (GCN)*, August 20, 2008, available from *www.gcn.com/online/vol1_no1/46946-1.html*.

6. Victor Oppleman, Oliver Friedrichs, and Brett Watson, "Chapter 7, Intrusion Detection and Prevention," *Extreme Exploits: Advanced Defenses Against Hardcore Hacks*, Emeryville, CA: McGraw-Hill/Osborne, 2005, available from *common.books24x7.com/book/id_11979/book.asp.*

7. Thomas H. Ptacek and Timothy N. Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," available from *insecure.org/stf/secnet_ids/secnet_ids.html*.

8. "Chapter XII, Deception in Cyber Attacks," in Lech J. Janczewski and Andrew M. Colarik, eds., *Cyber Warfare and Cyber Terrorism*, Hershey, PA: IGI Publishing, 2008, available from *common.books24x7.com/book/id_20791/book.asp*.

9. Wang Jiang, Anup K. Ghosh, and Huang Yih, "Internet Cleanroom: A System to Use On-Demand Virtualization to Enhance Client-Side Security," Washington, DC: Center for Secure Information Systems (CSIS), George Mason University, June 6, 2008.

10. Wang Jiang, Anup K. Ghosh, and Huang Yih, "Web Canaries: a Large-scale Distributed Sensor for Detecting Malicious Web Sites via a Virtualized Web Browser," Washington, DC: Center for Secure Information Systems (CSIS), George Mason University, November 2008, p. 6, available from *CollaborateCom.org/2008/program.php*.

CHAPTER 10

VOICE OVER IP: RISKS, THREATS, AND VULNERABILITIES*

Angelos D. Keromytis

INTRODUCTION

Voice over Internet Protocol (VoIP) and Internet Multimedia Subsystem (IMS) technologies are rapidly being adopted by consumers, enterprises, governments, and militaries. These technologies offer higher flexibility and more features than the traditional public-switched telephone network (PSTN) infrastructure, as well as the potential for lower cost through equipment consolidation and, for the consumer market, new business models. However, VoIP/IMS systems also represent a higher complexity in terms of architecture, protocols, and implementation, with a corresponding increase in the potential for misuse. Here, we begin to examine the current state of affairs on VoIP/IMS security through a survey of known/disclosed security vulnerabilities in bug-tracking databases. This chapter should serve as a starting point for understanding the threats and risks in a rapidly evolving set of technologies that are more frequently being deployed and used. Our goal is to gain a better understanding of the security landscape, with respect to VOIP/IMS, to encourage future research toward this and other similar emerging technologies.

The rate at which new technologies are being introduced and adopted by society has been steadily

---

*This work was supported by the French National Research Agency (ANR) under Contract ANR-08-VERS-017.

accelerating throughout human history. The advent of pervasive computing and telecommunications has reinforced this trend. In this environment of constant innovation, individuals, governments, and organizations have been struggling to manage the tension between reaping the benefits of new technologies while understanding and managing their risks. In this struggle, cost reductions, convenience, and new features typically overcome security concerns. As a result, security experts (but also the government and courts of law) are often left with the task of playing "catch up" with those who exploit flaws to further their own goals. This is the situation we find ourselves in with respect to one popular class of technologies collectively referred to as VoIP. VoIP, sometimes also referred to as Internet Multimedia Subsystem (IMS), refers to a class of products that enable advanced communication services over data networks. While voice is a key aspect in such products, video and other capabilities (e.g., collaborative editing, whiteboard file sharing, and calendaring) are all supported. The key advantages of VoIP/IMS are flexibility and low cost. The former derives from the generally open architectures and software-based implementation, while the latter is due to new business models, equipment, network-link consolidation, and ubiquitous consumer-grade broadband connectivity. Due to these benefits, VoIP has experienced a rapid uptake in both the enterprise and consumer markets. An increasing number of enterprises are replacing their internal phone switches with VoIP-based systems, both to introduce new features and to eliminate redundant equipment. Consumers have embraced a host of technologies with different features and costs, including Peer to Peer (P2P) calling, Internet-to-phone network bridging, and

wireless VoIP. These new technologies and business models are being promoted by a new generation of startup companies that are challenging the traditional status quo in telephony and personal telecommunications. As a result, a number of PSTN providers have already completed or are in the process of transitioning from circuit-switched networks to VoIP-friendly packet-switched backbones. Finally, as the commercial and consumer sectors go, so do governments and militaries due to cost reduction concerns and the general dependence on Commercial-off-the-Shelf (COTS) equipment for the majority of their computing needs. However, higher complexity is often the price we pay for more flexibility. In the case of VoIP/IMS technologies, a number of factors contribute to architectural, protocol, implementation, and operational complexity.

The number and complexity of the various features integrated in a product are perhaps the single largest source of complexity. For example, voice and video transmission typically allow for a variety of codecs which may be used in almost-arbitrary combinations. Since one of the biggest selling points for VoIP/IMS is feature-richness and the desire to unify personal communications under the same umbrella, this is a particularly pertinent concern.

Openness and modularity, generally considered desirable traits, allow for a number of independent implementations and products. Each of these comes with its own parameters and design choices. Interoperability concerns and customer feedback then lead to an ever-growing baseline of supported features for all products. A compounding factor to increasing complexity for much of the open VoIP is the "design-by-committee" syndrome, which typically leads to larger, more inclusive specifications than would otherwise

be the case (e.g., in a closed, proprietary environment such as the wire line telephony network from 20 years ago).

Because VoIP systems are envisioned to operate in a variety of environments, business settings, and network conditions, they must offer considerable configurability, which in turn leads to high complexity. Of particular concern are unforeseen feature interactions and other emergent properties. Finally, VoIPs are generally meant to work over a public data network (e.g., the Internet), or an enterprise/operator network that uses the same underlying technology. As a result, there is a substantial amount of non-VoIP infrastructure that is critical for the correct operation of the system, including such protocols/services as Dynamic Host Configuration Protocol (DHCP),[1] Domain Name System (DNS),[2] Trivial File Transfer Protocol/Bootstrap Protocol (TFTP/BOOTP),[3] Network Address Translation ([NAT],[4] and NAT traversal protocols such as Simple Traversal of UDP through NATs [STUN]),[5] Network Time Protocol (NTP),[6] Simple Network Management Protocol (SNMP),[7] routing the web (HTTP,[8] LS/SSL,[9] etc.), and many others. As we shall see, even a "perfectly secure" VoIP system can be compromised by subverting elements of this infrastructure. Because of this complexity, which manifests itself both in terms of configuration options and size of the code base for VoIP implementations, VoIP systems represent a very large attack surface. Thus, one should expect to encounter, over time, security problems arising from design flaws (e.g., exploitable protocol weaknesses), undesirable feature interactions (e.g., combinations of components that make new attacks possible or existing/known attacks easier), unforeseen dependencies (e.g., compromise paths through

seemingly unrelated protocols), weak configurations, and, not least, implementation flaws. In this chapter, we attempt a first effort at mapping out the space of VoIP threats and risks by conducting a survey of the "actually seen" vulnerabilities and attacks, as reported by the popular press and by bug-tracking databases. Our work is by necessity evolutionary in nature, and this chapter represents a current (and limited) snapshot of the complete space. Nonetheless, we believe that it will serve as a valuable starting point for understanding the bigger problem and as a basis for a more comprehensive analysis in the future.

Chapter Organization.

The remainder of this chapter is organized as follows. The second section contains a brief over view of two major VoIP technologies, Session Initiation Protocol (SIP) and Unlicensed Mobile Access (UMA). While we refer to other VoIP/IMS systems throughout the discussion, we focus on the specific two technologies as they are representative, widely used, and well-documented. We discuss VoIP threats in the third section, placing known attacks against VoIP systems within the taxonomy proposed by the VoIP Security Alliance. We analyze our findings in the fourth section. In the final section, we conclude with some preliminary thoughts on the current state of VoIP security, and on possible future directions for security research and practices.

VOIP TECHNOLOGIES OVERVIEW

In their simplest form, VoIP technologies enable two (or more) devices to transmit and receive real-time audio traffic that allows their respective users to communicate. In general, VoIP architectures are partitioned in two main components: signaling and media transfer. Signaling covers both abstract notions, such as endpoint naming and addressing, and concrete protocol functions such as parameter negotiation, access control, billing, proxying, and NAT traversal.

Depending on the architecture, quality of service (QoS) and device configuration/management may also be part of the signaling protocol (or protocol family). The media transfer aspect of VoIP systems generally includes a comparatively simpler protocol for encapsulating data, with support for multiple codecs and (often, but not always) content security. A commonly used media transfer protocol is Real-time Transport Protocol (RTP),[10] with a version supporting encryption and integrity, Secure Real-time Transport Protocol (SRTP),[11] defined but not yet widely used. The RTP protocol family also includes RTP Control Protocol (RTCP), which is used to control certain RTP parameters between communicating endpoints. However, a variety of other features are also generally desired by users and offered by providers as a means for differentiation by competing technologies and services, such as video, integration with calendaring, file sharing, and bridging to other networks (e.g., to the "regular" telephony network). Furthermore, a number of different decisions may be made when designing a VoIP system, reflecting different requirements and approaches to addressing, billing, mobility, security and access control, usability, and other

issues. Consequently, there exists a variety of different VoIP/IP Multimedia Subsystem (IMS) protocols and architectures. For concreteness, we will focus our attention on a popular and widely deployed technology: the Session Initiation Protocol (SIP).[12] We will also discuss the UMA architecture,[13] as a different approach to VoIP that is gaining traction among wireless telephony operators. In the rest of this section, we give a high-level overview of SIP and UMA, followed by a brief description of the salient points of a few other popular VoIP systems, such as H.323 and Skype. We will refer back to this overview in the third section of this chapter when we provide a discussion of the the threat and specific vulnerabilities.

Session Initiation Protocol (SIP).

SIP is a protocol standardized by the Internet Engineering Task Force (IETF) and is designed to support the setup of bidirectional communication sessions including, but not limited to, VoIP calls. It is similar in some ways to HTTP in that it is text-based, has a request-response structure, and even uses a mechanism based on the HTTP Digest Authentication[14] for user authentication. However, it is an inherently stateful protocol that supports interaction with multiple network components (e.g., middle boxes such as PSTN bridges). While its finite state machine is seemingly simple, in practice it has become quite large and complicated, an observation supported by the fact that the main SIP Requests for Comments (RFC) [15] is one of the longest ever defined. SIP can operate over a number of transport protocols, including Tranmission Control Protocol (TCP),[16] User Datagram Protocol (UDP),[17] and Stream Control Transmission Protocol (SCTP).[18] UDP

is generally the preferred method due to simplicity and performance, although TCP has the advantage of supporting Transport Layer Security (TLS) protection of call setup. However, recent work on Datagram TLS (DTLS)[19] may render this irrelevant. SCTP, on the other hand, offers several advantages over both TCP and UDP, including Denial of Service (DoS) resistance,[20] multi-homing and mobility support, and logical connection multiplexing over a single channel. In the SIP architecture, the main entities are endpoints (whether soft phones or physical devices), a proxy server, a registrar, a redirect server, and a location server. Figure 10.1 shows a high-level view of the SIP entity interactions.

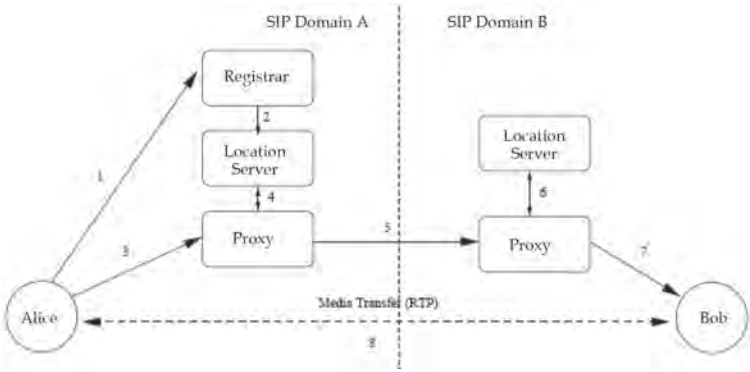


Figure 10.1. Session Initiation Protocol (SIP) Entity Interactions.

The User, Alice, registers with her domain's Registrar (1), which stores the information in the Location Server (2). When placing a call, Alice contacts her Local Proxy Server (3), which may consult the Location Server (4). A call may be forwarded to another Proxy

Server (5), which will consult its domain Location Server (6) before forwarding the call to the final recipient. After the SIP negotiation terminates, RTP is used directly between Alice and Bob to transfer media content. For simplicity, this diagram does not show the possible interaction between Alice and a Redirection Server (which would, in turn, interact with the Location Server). The registrar, proxy, and redirect servers may be combined, or they may be separate entities operated independently. Endpoints communicate with a registrar to indicate their presence. This information is stored in the location server. A user may be registered via multiple endpoints simultaneously. During call setup, the endpoint communicates with the proxy which uses the location server to determine where the call should be routed. This may be another endpoint in the same network (e.g., within the same enterprise), or another proxy server in another network. Alternatively, endpoints may use a redirect server to directly determine where a call should be directed and, redirect servers consult with the location server in the same way that proxy servers operate during call setup.

Once an end-to-end channel has been established (through one or more proxies) between the two endpoints, SIP negotiates the actual session parameters (such as the codecs, RTP ports, etc.) using the Session Description Protocol (SDP).[21] Figure 10.2 shows the message exchanges during a two-party call setup. Alice sends an INVITE message to the proxy server, optionally containing session parameter information encoded within SDP. The proxy forwards this message directly to Bob, if Alice and Bob are users of the same domain. If Bob is registered in a different domain, the message will be relayed to Bob's proxy, and from there to Bob. Note that the message may be forwarded

to multiple endpoints if Bob is registered from multiple locations. While these are ringing (or otherwise indicating that a call setup is being requested), RINGING messages are sent back to Alice. Once the call has been accepted, an OK message is sent to Alice containing Bob's preferred parameters encoded within SDP. Alice responds with an ACK message. Alice's session parameter preferences may be encoded in the INVITE or the ACK message. (See Figure 10.2.)

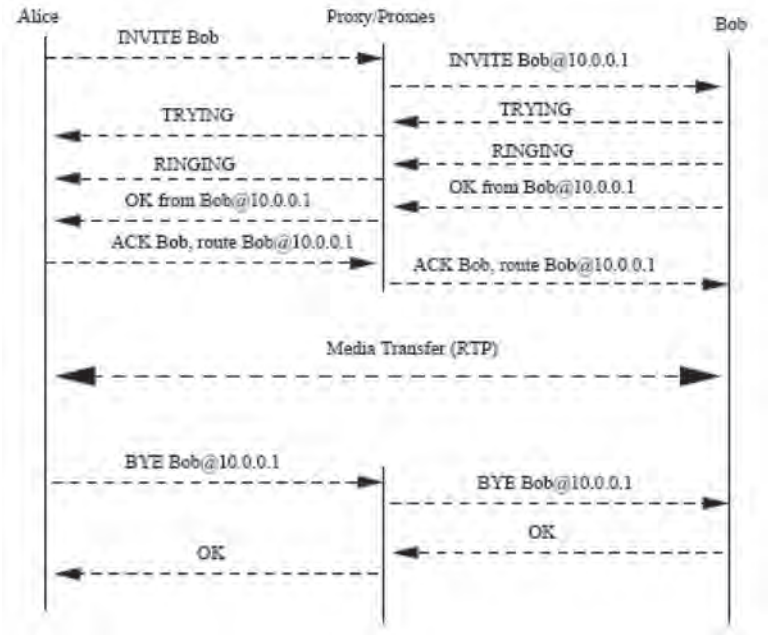


Figure 10.2. Message Exchanges During an SIP-Based Two-Party Call Setup.

Following this exchange, the two endpoints can begin transmitting voice, video, or other content (as negotiated) using the agreed-upon media transport protocol, typically RTP. While the signaling traffic

may be relayed through a number of SIP proxies, the media traffic is exchanged directly between the two endpoints. When bridging different networks, e.g., PSTN and SIP, media gateways may disrupt the end-to-end nature of the media transfer. These entities translate content (e.g., audio) between the formats that are supported by the different networks.

Because signaling and media transfer operate independent of each other, the endpoints are responsible for indicating to the proxies that the call has been terminated, using a BYE message which is relayed through the proxies along the same path as the call setup messages. There are many other protocol interactions supported by SIP that cover many common (and uncommon) scenarios including call forwarding (manual or automatic), conference calling, voicemail, etc. Typically, this is done by semantically overloading SIP messages such that they can play various roles in different parts of the call. The third section contains examples of how this flexibility and protocol modularity can be used to attack the system. All SIP traffic is transmitted over port 5060 (UDP or TCP). The ports used for the media traffic, however, are dynamic and negotiated via Session Description Protocol (SDP) during call setup. This poses some problems when NAT or firewalls are traversed. Typically, these have to be stateful and understand the SIP exchanges so that they can open the appropriate RTP ports for the media transfer. In the case of NAT traversal, endpoints may use protocols like STUN to enable communication. Alternatively, the Universal Plug-and-Play (uPnP) protocol 2 may be used in some environments, such as residential broadband networks consisting of a single subnet behind a NAT gateway. Authentication between endpoints, the registrar, and the proxy typically uses HTTP Digest Authentication, as shown in Figure

10.3. This is a simple challenge-response protocol that uses a shared secret key along with a username, domain name, a nonce, and specific fields from the SIP message to compute a cryptographic hash. Using this mechanism, passwords are not transmitted in plaintext form over the network. It is worth noting that authentication may be requested at almost any point.

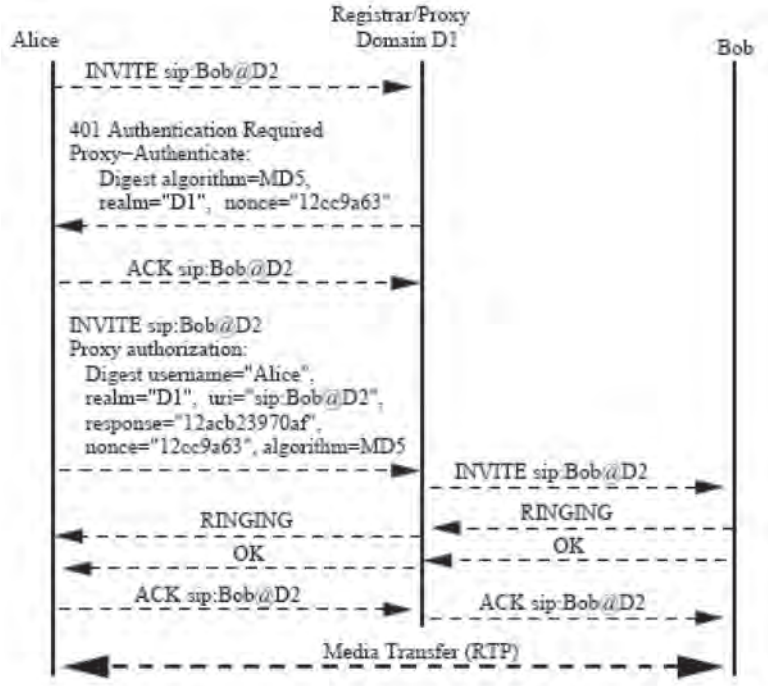


Figure 10.3. SIP Digest Authentication.

Later, we will see an example where this protocol can be abused by a malicious party to conduct toll fraud in some environments. For more complex authentication scenarios, SIP can use Secure/Multipurpose Internet Mail Extensions (S/MIME) encapsulation[22] to carry complex payloads, including public keys and certificates. When TCP is used as the transport protocol for SIP, TLS can be used to protect the SIP

messages. TLS is required for communication among proxies, registrars, and redirect servers, but only recommended between endpoints and proxies or registrars. Alternatively, IPsec[23] may be used to protect all communications, regardless of the transport protocol. However, because few implementations integrate SIP, RTP, and IPsec, it is left to system administrators to figure out how to setup and manage such configurations.

Unlicensed Mobile Access.

UMA is a 3 Generation Partnership Project (3GPP) standard for enabling transparent access to mobile circuit-switched voice networks, packet-switch data networks, and IMS services using any IP-based substrate. Handsets supporting UMA can roam between the operator's wireless network (usually referred to as a Radio Access Network, or RAN) and the Internet without losing access. For example, a call that is initiated over the RAN can then be routed without being dropped and with no user intervention over the public Internet if conditions are more favorable (e.g., stronger WiFi signal in the user's premises, or in a hotel wireless hotspot while traveling abroad). For consumers, UMA offers better connectivity and the possibility of lower cost by enabling new business models and reducing roaming charges (under some scenarios). For operators, UMA reduces the need for additional spectrum; cell phone towers and related equipment. A variety of cell phones supporting UMA over WiFi currently exist, along with home gateways and USB-stick soft phones. More recently, some operators have introduced femto cells (ultra-low power RAN cells intended for consumer-directed deployment)

that can act as UMA gateways, allowing any mobile handset to take advantage of UMA where such devices are deployed. The basic approach behind UMA is to encapsulate complete Global System for Mobile (GSM) and 3rd Generation (3G) radio frames (except for the over-the-air crypto) inside IP packets. These can then be transmitted over any IP network, including the Internet. This means that the mobile operator can continue to use the existing back-end equipment; all that is needed is a gateway that encapsulates the GSM/3G frames and injects them to the existing circuit-switched network (for voice calls), as can be seen in Figure 10.4. To protect both signaling and media traffic confidentiality and integrity while traversing un-trusted (and untrustworthy) networks, UMA uses Internet Protocol Security (IPSec). All traffic between the handset (or, more generally, UMA endpoint) and the provider's UMA Network Controller (or a firewall/Virtual Private Network [VPN] concentrator screening traffic) is encrypted and integrity-protected using Encapsulating Security Payload (ESP).[24]

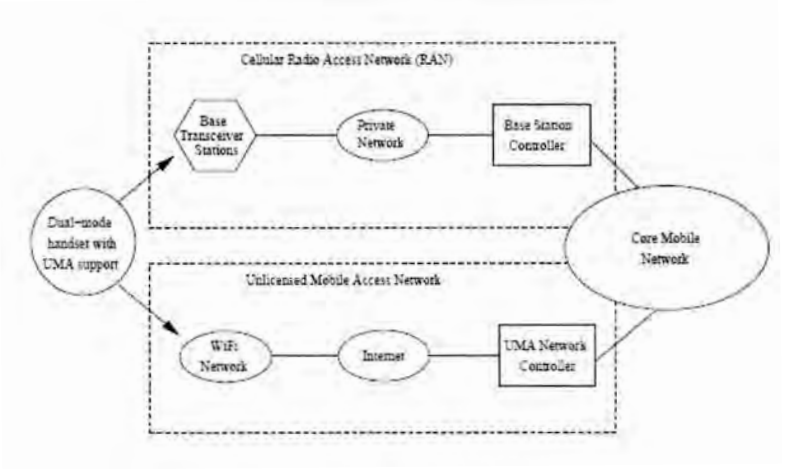


Figure 10.4. Unlicensed Mobile Access (UMA) Conceptual Architecture During a Call Setup.

The use of IPSec provides a high level of security for network traffic, once keys and other parameters have been negotiated. For that purpose, the Internet Key Exchange version 2 (IKEv2) key management protocol[25] is used. Authentication uses the Extensible Authentication Protocol method for GSM Subscriber Identity Module (EAP-SIM)[26] (for GSM handsets) and Extensible Authentication Protocol method for UMTS Authentication and Key Agreement (EAP-AKA)[27] (for UMTS handsets) profiles. Authentication is asymmetric: the provider authenticates to the handset using digital signatures and public key certificates, while the handset authenticates using a SIM-embedded secret key. It is worth pointing out that UMA provides stronger authentication guarantees than the baseline cell phone network in that the provider does not authenticate to the handset in a RAN. Furthermore, the cryptographic algorithms used in IPSec Advanced Encryption Standard and 3 Data Encryption Standard (AES and 3DES) are considered significantly stronger than the on-the-air algorithms used in GSM. Despite the use of strong cryptography and sound protocols, UMA introduces some new risks in the operator networks, since these now have to be connected to the public Internet in a much more intimate fashion. In particular, the security gateway must process IPSec traffic, including the relatively complex IKEv2 protocol, and a number of UMA-related discovery and configuration protocols. These significantly increase the attack surface and overall security exposure of the operators.

Other VoIP/IMS Systems.

H.323 is an ITU defined protocol family for VoIP (audio and video) over packet-switched data networks. The various sub protocols are encoded in Abstract Syntax Notation One (ASN.1) format. In the H.323 world, the main entities are terminals (software or physical phones), a gateway, a gatekeeper, and a back-end service. The gate keeper is responsible for address resolution, controlling bandwidth use, and other management functions while the gateway connects the H.323 network with other networks (e.g., PSTN, or a SIP network). The back-end service maintains data about the terminals, including configuration, access billing rights, etc. An optional multipoint control unit may also exist to enable multipoint communications, such as a teleconference. To setup an H.323 call, terminals first interact with the gatekeeper using the H.225 protocol over either TCP or UDP to receive authorization and to perform address resolution. Using the same protocol, they then establish the end-to-end connection to the remote terminal (possibly through one or more gateways). At that point, H.245 over TCP is used to negotiate the parameters for the actual media transfer, including ports, which uses RTP (as in the case of SIP). Authentication may be requested at several steps during call setup, and typically depends on symmetric keys but may also use digital signatures. Voice encryption is also supported through SRTP and MIKEY.[28] Unlike SIP, H.323 does not use a well-known port, making firewall traversal even more complicated. Skype3 is a P2P VoIP system that was originally available as a soft phone for desktop computers but has since been integrated into cell phones and other handheld devices, either as an add-on or

as the exclusive communication mechanism. It offers voice, video, and text messaging to all other Skype users free of charge, and provides bridging (typically for a fee) to the PSTN both for outgoing and incoming calls and text messages (SMS). The underlying protocol is proprietary, and the software itself incorporates several anti-reverse engineering techniques. Nonetheless, some analysis[29] and reverse engineering[30] have taken place, indicating both the ubiquitous use of strong cryptography and the presence of some software bugs (at the time of the work). The system uses a centralized login server but is otherwise fully distributed with respect to intra-Skype communications. A number of chat (IM) networks, such as the AOL Instant Messenger, Microsoft's Live Messenger, Yahoo! Messenger, and Google Talk offer voice and video capabilities as well. Although each network uses its own (often proprietary) protocol, bridges exist between most of them, allowing inter-IM communication at the text level. In most of these networks, users can place outgoing voice calls to the PSTN. Some popular IM clients also integrate SIP support.

VOIP THREATS

In trying to understand the threat space against VoIP, our approach is to place known vulnerabilities within a structured framework. While a single taxonomy is not likely to be definitive, using several different viewpoints and mapping the vulnerability space along several axes may reveal trends and other areas that merit further analysis. As a starting point, we use the taxonomy provided by the Voice over IP Security Alliance (VOIPSA). VOIPSA is a vendor-neutral, not for profit organization composed of VoIP and security

vendors, organizations, and individuals with an interest in securing VoIP protocols, products, and installations. In addition, we place the surveyed vulnerabilities within the traditional threat spa of confidentiality, integrity, and availability (CIA). Finally, we ascertain whether the vulnerabilities exploit bugs in the protocol, implementation, or system configuration. In future work, we hope to expand the number of views to the surveyed vulnerabilities and to provide more in-depth analysis. The VOIPSA security threat taxonomy[31] aims to define the security threats against VoIP deployments, services, and end users. The key elements of this taxonomy are:

1. Social threats are aimed directly against humans. For example, misconfigurations, bugs, or bad protocol interactions in VoIP systems may enable or facilitate attacks that misrepresent the identity of malicious parties to users. Such attacks may then act as stepping stones for further attacks such as phishing, theft of service, or unwanted contact (spam).

2. Eavesdropping, interception, and modification threats cover situations where an adversary can unlawfully and without authorization from the parties concerned listen in on the signaling (call setup) or the content of a VoIP session, and possibly modify aspects of that session while avoiding detection. Examples of such attacks include call re-routing and interception of unencrypted RTP sessions.

3. Denial of service (DoS) threats have the potential to deny users access to VoIP services. This may be particularly problematic in the case of emergencies, or when a DoS attack affects all of a user's or an organization's communication capabilities (i.e., when all VoIP and data communications are multiplexed over the same network which can be targeted through a DoS

attack). Such attacks may be VoIP-specific (exploiting flaws in the call setup or the implementation of services), or VoIP-agnostic (e.g., generic traffic flooding attacks). They may also involve attacks with physical components (e.g., physically disconnecting or severing a cable) or through computing or other infrastructures (e.g., disabling the DNS server, or shutting down power).

4. Service abuse threats covers the improper use of VoIP services, especially (but not exclusively) in those situations where such services are offered in a commercial setting. Examples of such threats include toll fraud and billing avoidance.[32]

5. Physical access threats refer to inappropriate/unauthorized physical access to VoIP equipment, or to the physical layer of the network (following the ISO 7-layer network stack model).

6. Interruption of services threats refer to nonintentional problems that may nonetheless cause VoIP services to become unusable or inaccessible. Examples of such threats include loss of power due to inclement weather, resource exhaustion due to oversubscription, and performance issues that degrade call quality.

In our discussion of vulnerabilities (whether theoretical or demonstrated) that follows, we shall mark each item with a tuple (V, T, K), where: V ∈ {1, 2, 3, 4, 5, 6}, where each number refers to an element in the VOIPSA threat taxonomy from above; T ∈ {C1 , I1 , A1}, referring to confidentiality, integrity and availability, respectively; K ∈ {P2 , I2 , C2 }, referring to protocol, implementation, and configuration respectively; and confidentiality via a configuration problem or bug. In some cases, the same underlying vulnerability may be used to perform different types of attacks. We will be discussing all such significant attack variants.

Disclosed Vulnerabilities.

Threats against VoIP system availability that exploit implementation weaknesses are fairly common. For example, some implementations were shown to be vulnerable to crashes or hanging (live clock) when given empty, malformed, or large volumes of[33] INVITE or other messages (3, A1, I2). It is worth noting that the same vulnerability may be present across similar protocols on the same platform and product[34] due to code sharing and internal software structure, or to systems that need to understand VoIP protocols but are not nominally part of a VoIP system.[35] The reason for the disproportionately large number of denial of service vulnerabilities is because of the ease with which such failure can be diagnosed, especially when the bug is discovered through automated testing tools (e.g., fuzzers). Many of these vulnerabilities may in fact be more serious than a simple denial of service due to a crash, and could possibly lead to remote code injection and execution. Unexpected interactions between different technologies used in VoIP systems can also lead to vulnerabilities. For example, in some cases cross-site scripting (XSS) attacks were demonstrated against the administrator- and customer-facing management interface (which was web-based) by injecting malicious Java script in selected SIP messages[36] (1, I1, I2), often through Structured Query Language (SQL) injection vulnerabilities.[37] The same vulnerability could also be used to commit toll fraud by targeting the underlying database (4, I1, I2). XSS attacks that are not web-oriented have also been demonstrated, with one of the oldest VoIP-related vulnerabilities[38] permitting shell command execution. Another web-oriented attack

vector is Cross Site Request Forgery (CSRF), whereby users visiting a malicious page can be induced to automatically (without user intervention, and often without any observable indications) perform some action on the web servers (in this case, VoIP web-based management interface) that the browser is already authenticated to).[39] Other privilege-escalation vulnerabilities through the web interface also exist.[40] The complexity of the SIP finite state machine has sometimes led to poor implementations. For example, one vulnerability[41] allowed attackers to confuse a phone receiving a call into silently completing the call, which allowed the adversary to eavesdrop on the device's surroundings.

The same vulnerability could be used to deny call reception of the target, since the device was already marked as busy. In other cases, it is unclear to developers what use of a specific protocol field may be, in which case they may silently ignore it. Occasionally, such information is critical for the security of the protocol exchange, and omitting or not checking it allows adversaries to perform attacks such as man-in-the-middle, or traffic interception,[42] or bypass authentication checks.[43]

Since SIP devices are primarily software-driven, they are vulnerable to the same classes of vulnerabilities as other software. For example, buffer overflows are possible even against SIP "headphones," much less soft phones, allowing adversaries to gain complete control of the device.[44] Such vulnerabilities typically arise from a combination of poor (nondefensive) programming practices, insufficient testing, and the use of languages, such as C and C++ that support unsafe operations. Sometimes, these vulnerabilities appear in software that is not directly used in VoIP but must be

VoIP-aware, e.g., fire walls[45] or protocol analyzers.[46] It is also worth noting that these are not the only types of vulnerabilities that can lead to remote code execution.[47] Other input validation failures can allow attackers to download arbitrary files from a user's machine (1, C1, I2) or to place calls[48] (1, I1, I2) by supplying specially encoded URIs[49] or other parameters. A significant risk with VoIP devices is the ability of adversaries to misrepresent their identity (e.g., their calling number). Such vulnerabilities[50] sometimes arise due to the lack of cross-checking of information provided across several messages during call setup and throughout the session (1, I1, I2).

Similar failures to crosscheck and validate information can lead to other attacks, such as indicating whether there is pending voicemail for the user[51] (1, I1, I2), or where attackers may spoof incoming calls by directly connecting to a VoIP phone[52] (1, I1, I2).

Undocumented, on-by-default features are another source of vulnerabilities. These are often remnants from testing and debugging during development that were not disabled when a product shipped.[53] As a result, they often offer privileged access to services and data on a device that would not otherwise be available[54] (1, C1, I2). One particularly interesting vulnerability allowed an attacker to place outgoing calls through the web management interface[55] (4, I1, C2). A significant class of vulnerabilities in VoIP devices revolves around default configurations, and in particular default usernames and passwords[56] (2, C1 + I1, C2). Lists of default accounts are easy to find on the Internet via a search engine. Users often do not change these settings; this seems to be particularly so for administrative accounts, which are rarely (if ever) used in the home/Small Office Home Office (SOHO)

environment. Other default settings involve Network Time Protocol (NTP) servers[57] and DNS servers[58] (2, C1 + I1, C2). Since the boot and VoIP stacks are not necessarily tightly integrated, interaction with one protocol can have adverse effects (e.g., changing the perceived location of the phone) in the other protocol[59] [2, C1, I2]). Other instances of such vulnerabilities involve improper/insufficient credential checking by the registrar or proxy[60] or by the SNMP server,[61] which can lead to traffic interception (2, C1, I2) and user impersonation (1, I1, I2). The integration of several capabilities in VoIP products, e.g., a web server used for the management interface, can lead to vulnerabilities being imported to the VoIP environment that would not otherwise apply. In the specific example of an integrated web server, directory traversal bugs[62] or similar problems (such as lack of proper authentication in the web interface)[63] can allow adversaries to read arbitrary files or other information from the device (1, C1, I2). SIP (or, more generally, VoIP) components integrated with firewalls may also interact in undesirable ways. For example, improper handling of registration requests may allow attackers to receive messages intended for other users[64] (2, C1, I2). Other such examples include failure to authenticate server certificates in wireless environments, enabling man-in-the-middle and eavesdropping attacks[65] (2, C1, I2).

Predictability and lack of proper use (or sources) of randomness is another vulnerability seen in VoIP products. For example, predictable values in SIP header messages[66] allows malicious users to avoid registering, but continue using the service (4, I1, I2). Protocol responses to carefully crafted messages can reveal information about the system or its users to an attacker. Although this has been long understood in limited-

domain protocols (e.g., remote login), with measures taken to normalize responses such that no information is leaked, the complexity of VoIP (and other) protocols make this infeasible. As a result, information disclosure vulnerabilities abound[67] (1, C1, I2).

Some of the most serious nonimplementation type of vulnerabilities are those where the specification permits behavior that is exploitable. For example, certain vendors permit the actual Uniform Resource Indentifier (URI) in a SIP INVITE call and the URI used as part of the Digest Authentication to differ, which (while arguably permitted by the specification) allows credential reuse and toll fraud[68] (4, I1, P2). While rare, protocol-level vulnerabilities also exist. These represent either outright bugs in the specification, or unseen interaction between different protocols or protocol components. For large, complicated protocols such as SIP and H.323, where components (code, messages, etc.) are semantically overloaded and reused, it is perhaps not surprising that such emergent properties exist. One good example is the relay attack that is possible with the SIP Digest Authentication,[69] whereby an adversary can reuse another party's credentials to obtain unauthorized access to SIP or PSTN services (such as calling a premium or international phone line) (4, I1, P2). This attack is possible because in an authentication attack, both depicted in Figure 10.5, an authentication may be requested in response to an INVITE message that is not usable in, for example, placing fraudulent calls.

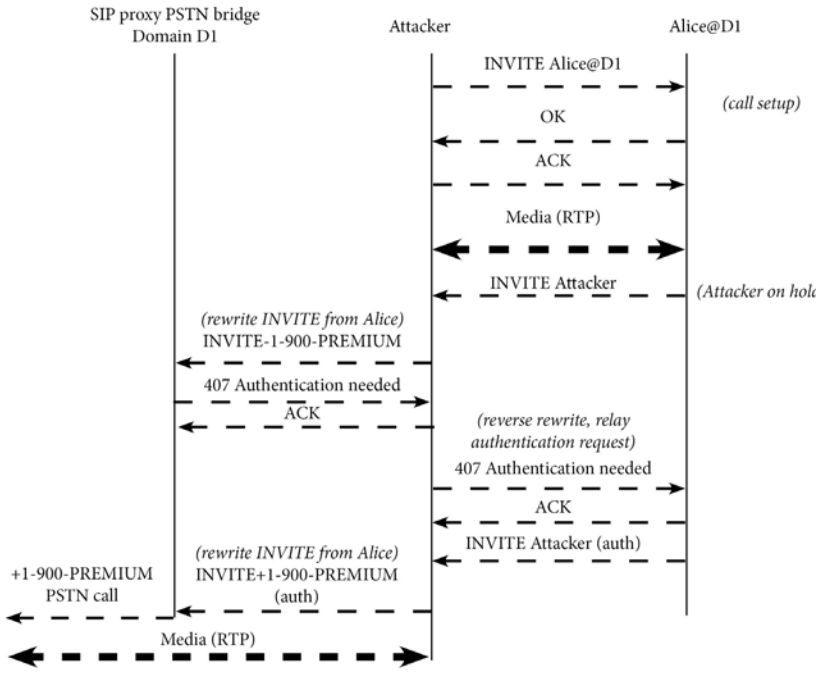


Figure 10.5. SIP Relay Attack.

DISCUSSION

Looking at the vulnerabilities we have considered, a few patterns emerge. First, as we can see in our informal classification of vulnerability effects shown in Figure 10.6, half of the problems lead to a denial of service in either an end-device (phone or soft phone) or a server (proxy, registrar, etc.). This is not altogether surprising, since denial of service (especially a crash) is something that is easily diagnosed. In many cases, the problem was discovered by automated testing, such as protocol or software fuzzing; software failures are relatively easy to determine in such settings. Some of these vulnerabilities could in

fact turn out to be more serious, e.g., a memory corruption leading to a crash could be exploitable to mount a code injection attack. The second largest class of vulnerabilities allows an adversary to control the device, whether by code injection, default passwords and services, or authentication failures. Note that we counted a few of the vulnerabilities (approximately 10 percent) more than once in this classification. The same pattern with respect to the predominance of denial of service vulnerabilities holds when we look at the breakdown according to the VOIPSA taxonomy, shown in Figure 10.7. It should not be surprising that, given the nature of the vulnerabilities disclosed in Common Vulnerabilities and Exposures (CVE), we have no data on physical access and (accidental) interruption of services vulnerabilities. Furthermore, while "Access to Services" was a non-negligible component in the previous breakdown, it represents only 4 percent here. The reason for this apparent discrepancy is in the different definitions of service: the specific element in the VOIPSA taxonomy refers to VoIP-specific abuse, whereas our informal definition covers lower-level system components which may not be usable in, for example, placing fraudulent calls. One state (data) resident on the system falls into the "access to data" category. The other observation here, is that while the VOIPSA taxonomy covers a broad spectrum of concerns or VoIP system designers and operators, its categories are perhaps too broad (and, in some cases, imprecise) to help with characterizing the types of bugs we have examined.

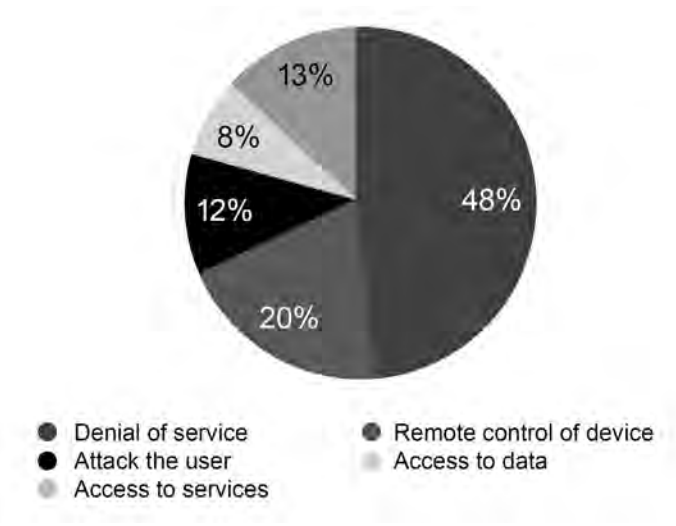


Figure 10.6. Vulnerability Breakdown Based on Effect.

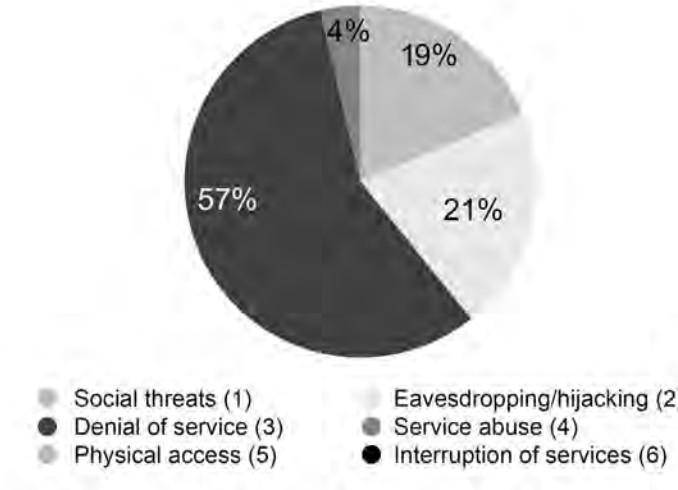


Figure 10.7. Vulnerability Breakdown Based on VOIPSA Taxonomy.

Most categories are self explanatory; "attack the user" refers to vulnerabilities that permit attackers to affect the user/administrator of a device without necessarily compromising the system or getting access to its data or services. XSS attacks and traffic eavesdropping attacks fall in this category, whereas attacks that compromise state (data) resident on the system fall in the "access to data" category.

The vulnerability breakdown according to the traditional CIA security concerns again reflects the predominance of denial of service threats against VoIP systems, as seen in Figure 10.8. However, we can see that integrity violations (e.g., system compromise) are a sizable component of the threat space, while confidentiality violations are seen in only 15 percent of disclosed vulnerabilities. This represents an inversion of the perceived threats by users and administrators, who (anecdotal evidence suggests) typically worry about such issues as call interception and eavesdropping. Finally, Figure 10.9 shows the breakdown based on source of vulnerability. The overwhelming majority of reported problems arise from implementation issues, which should not be surprising given the nature of bug disclosure. Problems arising from configuration represented 7 percent of the total space, including such items as privileged services left turned on and default username/passwords. However, note that the true picture (i.e., what actually happens with deployed systems) is probably different in that configuration problems are most likely undercounted: such problems are often site-specific and are not reported to bug-disclosure databases when discovered. On the other hand, implementation and protocol problems are prime candidates for disclosure. What is surprising is the presence of protocol vulnerabilities; one would expect that such problems were

discovered and issued during protocol development, specification, and standardization. Their mere existence potentially indicates high protocol complexity. The vulnerability analysis contained in this chapter is, by its nature, static: we have presented a snapshot of known problems with VoIP systems, with no correlation with (and knowledge of) actual attacks exploiting these, or other vulnerabilities. A complete analysis of the threat space would also contain a dynamic component, whereby attacker behavior patterns and trends would be analyzed vis-à-vis actual, deployed VoIP systems or, lacking access to such, simulacra thereof.[70]

Figure 10.8. Vulnerability Breakdown Based on Source (I2, C2, P2).

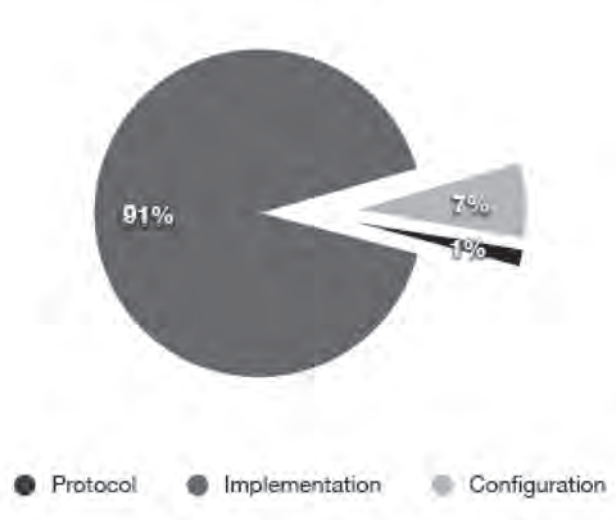


Figure 10.9. Vulnerability Breakdown Based on Source (I2, C2, P2).

CONCLUSIONS

We can draw some preliminary conclusions with respect to threats and potential areas for future research based on the data examined so far. These can be summarized as follows:

1. The large majority of disclosed threats focused on denial of services attacks are based on implementation issues. While fault-tolerance techniques can be applied in the case of servers (replication, hot standby, Byzantine fault tolerance, etc.), it is less clear how to provide similar levels of protection at acceptable cost and usability to end user devices. Unfortunately, the ease with which mass DoS attacks can be launched over the network against client devices means that they represent an attractive venue for attackers to achieve the same impact.

2. Code injection attacks in their various forms (buffer overflow, cross-site scripting, SQL injection, etc.) remain a problem. While a number of techniques have been developed, we need to do a better job at deploying and using them where possible, and devising new techniques suitable for the constrained environments that some vulnerable VoIP devices represent.

3. Weak default configurations remain a problem, as they do across a large class of consumer and enterprise products and software. The situation is likely to be much worse in the real world, considering the complexity of securely configuring a system with as many components as VoIP. Vendors must make an effort to provide secure-by-default configurations, and to educate users on how to best protect their systems. Administrators are in need of tools to analyze their existing configurations for vulnerabilities. While there are some tools that dynamically test network components (e.g., firewalls), we need tools that work higher in the protocol and application stack (i.e., interacting at the user level). Furthermore, we need ways of validating configurations across multiple components and protocols.

4. Finally, there is simply no excuse for protocol-level vulnerabilities. While techniques exist for analyzing and verifying security protocols, they do not seem to cope well with complexity. Aside from using such tools and continuing their development, protocol designers and standardization committees must consider the impact of their decisions on system implementers, i.e., whether it is likely that a feature or aspect of the protocol is likely to be misunderstood and/or misimplemented. Simpler protocols are also desirable, but seem incompatible with the trends we have observed in standardization bodies. Our plans for future work include expanding the data set we used for

our analysis to include findings from academic work, adding and presenting more views (classifications) to the data, and developing dynamic views to VoIP-related misbehavior.

ENDNOTES - CHAPTER 10

1. R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), March 1997, Updated by RFCs 3396, 4361, 5494.

2. P. V. Mockapetris, "Domain Names—Concepts and Facilities," RFC 1034 (Standard), November 1987, Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592; P. V. Mockapetris, "Domain Names—Implementation and Specification," RFC 1035 (Standard), November 1987, Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343.

3. K. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350 (Standard), July 1992, Updated by RFCs 1782, 1783, 1784, 1785, 2347, 2348, 2349; R. Finlayson, "Bootstrap loading using TFTP," RFC 906, June 1984.

4. P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022 (Informational), January 2001.

5. J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389 (Proposed Standard), October 2008.

6. D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis," RFC 1305 (Draft Standard), March 1992.

7. D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411 (Standard), December 2002, Updated by RFC 5343.

8. T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0," RFC 1945 (Informational), May 1996; R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616 (Draft Standard), June 1999, Updated by RFC 2817.

9. T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), August 2008.

10. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), July 2003, Updated by RFC 5506.

11. I. Johansson and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences," RFC 5506 (Proposed Standard), April 2009.

12. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), June 2002, Updated by RFCs 3265, 3853, 4320, 4916, 5393.

13. 3GPP, "Generic Access Network," available from *www.3gpp.org/ftp/Specs/html-info/43318.htm, 2009*.

14. J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), June 1999.

15. Rosenberg, Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley, and Schooler, "SIP: Session Initiation Protocol."

16. J. Postel, "Transmission Control Protocol," RFC 793 (Standard), September 1981, Updated by RFCs 1122, 3168.

17. J. Postel, "User Datagram Protocol," RFC 768 (Standard), August 1980.

18. L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286 (Informational), May 2002.

19. E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," RFC 4347 (Proposed Standard), April 2006.

20. M. Handley, E. Rescorla, and IAB, "Internet Denial-of-Service Considerations," RFC 4732 (Informational), December 2006.

21. M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566 (Proposed Standard), July 2006.

22. B. Ramsdell, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification," RFC 3851 (Proposed Standard), July 2004.

23. S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Standard), December 2005.

24. S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303 (Proposed Standard), December 2005.

25.C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," RFC 4306 (Proposed Standard), December 2005, Updated by RFC 5282.

26. H. Haverinen and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186 (Informational), January 2006.

27. J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," RFC 4187 (Informational), January 2006.

28. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," RFC 3830 (Proposed Standard), August 2004, Updated by RFC 4738.

29. Tom Berson, "Skype Security Evaluation," *Tech. Rep*., October 2005; S. A. Baset and H. Schulzrinne, "An Analysis of

the Skype Peer-to-Peer Telephony Protocol," in *Proceedings of INFOCOM*, April 2006.

30. P. Biondi and F. Desclaux, "Silver Needle in the Skype," in BlackHat Europe Conference, March 2006, available from *www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf*.

31. VoIP Security Alliance, "VoIP Security and Privacy Threat Taxonomy, version 1.0," October 2005, available from *www.voipsa.org/Activities/taxonomy.php*.

32. "Two charged with VoIP fraud," The Register, June 2006, available from *www.theregister.co.uk/2006/06/08/voip fraudsters nabbed/*; "Fugitive VOIP hacker cuffed in Mexico," The Register, February 2009, available from *www.theregister.co.uk/2009/02/11/fugitive voip hacker arrested/*.

33. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2007-4753; CVE-2007-0431; CVE-2007-4553; CVE-2003-1114; CVE-2006-1973; CVE-2007-0648; CVE-2007-2270; CVE-2007-4291, 4292; CVE-2008-3799, 3800, 3801, and 3802; CVE-2009-1158; CVE-2004-0054; CVE-2001-0546; CVE-2002-2266; CVE-2004-0498; CVE-2004-2344; CVE-2004-2629; CVE-2004-2758; CVE-2007-4429; CVE-2006-5084; CVE-2005-3267; CVE-2004-1777; CVE-2003-1108-1113; CVE-2003-1115; CVE-2004-0504; CVE-2005-4466; CVE-2006-5445; CVE-2006-2924; CVE-2006-0739, 0738, and 0737; CVE-2007-6371; CVE-2007-5583; CVE-2007-5537; CVE-2007-4924; CVE-2007-4459; CVE-2007-4455; CVE-2007-4382; CVE-2007-4366; CVE-2007-3441-3445; CVE-2007-3436 and 3437; CVE-2007-3369, 3368; CVE-2007-3361-3363; CVE-2007-3348-3351; CVE-2007-3322 and 3321; CVE-2007-3318 and 3317; CVE-2007-2297; CVE-2007-1693; CVE-2007-1650; CVE-2007-1594; CVE-2007-1590; CVE-2007-1561; CVE-2007-1542; CVE-2007-1306; CVE-2007-0961; CVE-2008-0095; CVE-2008-0263; CVE-2008-1249; CVE-2008-1741; CVE-2008-1745; CVE-2008-1747 and 1748; CVE-2008-1959; CVE-2008-2119; CVE-2008-2732; CVE-2008-2733; CVE-2008-2734 and 2735; CVE-2008-3157; CVE-2008-3210; CVE-2008-3778; CVE-2008-4444; CVE-2008-5180; CVE-2008-6140; CVE-2008-6574 and 6575; CVE-2009-0871; CVE-2009-0636; CVE-2009-0630; CVE-2009-0631; CVE-2007-5591; CVE-2007-5556; CVE-2007-5369; CVE-2007-2886; CVE-2006-7121; CVE-2006-6411;

CVE-2006-5233; CVE-2006-5231; CVE-2005-3989; CVE-2004-1977; CVE-2002-0882; CVE-2002-0880; CVE-2002-0835;

34. "CVE-2007-4291," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4291*.

35. "CVE-2005-4464," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-4464*.

36. "CVE-2007-5488," 2007 available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5488*. CVE-2007-5411, CVE-2008-0582, CVE-2008-0583, CVE-2008-0454, CVE-2006-2925, CVE-2007-2191.

37. "CVE-2008-6509," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6509*. "CVE-2008-6573," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6573*.

38. "CVE-1999-0938," 1999, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0938*.

39. "CVE-2008-1250," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1250*.

40. "CVE-2008-6708," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6708*.

41. "CVE-2007-4498," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4498*.

42. "CVE-2007-3319," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3319*.

43. "CVE-2007-3177," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3177*, 2007. "CVE-2007-0334," available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0334*.

44. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2003-1114, CVE-2003-1110, CVE-2003-1111, CVE-2005-4050, CVE-2007-4294, CVE-2007-4295, CVE-2004-0056, CVE-2004-0117, CVE-2005-3265, CVE-2004-1114, CVE-2003-0761, CVE-2006-4029, CVE-2006-3594,

CVE-2006-3524, CVE-2006-0359, CVE-2006-0189, CVE-2007-5788, CVE-2007-3438, CVE-2007-2293, CVE-2007-0746, CVE-2008-0528, CVE-2008-0530, CVE-2008-0531, CVE-2008-2085, CVE-2007-4489, CVE-2005-2081.

45. "CVE-2003-0819," 2003, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0819*.

46. "CVE-2005-1461," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1461*.

47. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2006-5084, CVE-2008-2545, CVE-2008-1805, CVE-2007-5989, CVE-2007-3896, CVE-2008-6709.

48. "CVE-2008-1334," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1334*.

49. "CVE-2006-2312," 2006, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2312*.

50. "CVE-2005-2181," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2181*.

51. "CVE-2005-2182," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2182*.

52. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2007-5791, CVE-2007-3347, CVE-2007-3320.

53. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2006-0305, CVE-2006-0302, CVE-2005-3804, CVE-2005-3724, CVE-2005-3715.

54. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2006-0360, CVE-2007-3439, CVE-2006-0374, CVE-2005-3723, CVE-2005-3721, CVE-2005-3718.

55. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2007-3440, CVE-2008-1248.

56. Examples of Common Vulnerabilities and Exposures are available from: *cve.mitre.org/cve/index.html*; CVE-2005-3718, CVE-2006-5038, CVE-2008-4874, CVE-2007-3047, CVE-2008-1334, CVE-2006-0834, CVE-2005-3803, CVE-2005-3719, CVE-2005-3717, CVE-2005-3716, CVE-2005-0745, CVE-2002-0881.

57. "CVE-2006-0375," 2006, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0375*.

58. "CVE-2005-3725," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3725*.

59. "CVE-2007-5361," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5361*.

60. "CVE-2008-5871," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-5871*.

61. "CVE-2005-3722," 2005, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3722*.

62. "CVE-2008-4875," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4875*.

63. "CVE-2008-6706," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6706*, 2008. "CVE-2008-6707," available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6707*.

64. "CVE-2007-6095," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6095*.

65. "CVE-2008-1114," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1114*, 2008. "CVE-2008-1113," available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1113*.

66. "CVE-2002-1935," 2002, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1935*.

67. "CVE-2006-4032," available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4032*, 2006. "CVE-2008-3903," 2008, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3903*.

68. "CVE-2007-5469," available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5469*, 2007. "CVE-2007-5468," 2007, available from *cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5468*.

69. R. State, O. Festor, H. Abdelanur, V. Pascual, J. Kuthan, R. Coeffic, J. Janak, and J. Loroiu, "SIP Digest Authentication Relay Attack," March 2009, available from *tools.ietf.org/html/draft-state-sip-relay-attack-00*.

70. M. Nassar, R. State, and O. Festor, "VoIP Honeypot Architecture," in Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, May 2007, pp. 109–118.

CHAPTER 11

TOWARD FOOLPROOF IP NETWORK CONFIGURATION ASSESSMENTS*

Rajesh Talpade

INTRODUCTION

Internet protocol (IP) networks have come of age. They are increasingly replacing leased-line data infrastructure and traditional phone service, and are expected to offer Public Switched Telephone Network (PSTN) - quality service at a much lower cost. As a result, there is an urgent interest in ensuring IP network security, reliability, and quality of service (QoS). In fact, regulators are now requiring compliance with IP-related mandates. This chapter discusses the complex nature of IP networks and how that complexity makes them particularly vulnerable to faults and intrusions. It describes regulatory efforts to mandate assessment, explains why many current approaches to IP assessment fall short, and describes the requirements for an effective solution to satisfy business, government, and regulatory requirements.

IP networks throughout the public and private sectors are now mainstream. Everyday, IP networks are responsible for transporting real-time and critical voice, video, and data traffic. As a result, it is no longer acceptable for IP networks to deliver "best-effort" service. They are expected to perform at carrier-grade level. However, it is enormously challenging to deploy

---

* This work was supported in part by the U.S. Department of Homeland Security Science & Technology Directorate under Contract No. NBCHC050092.

IP networks and assure consistent, and high quality service delivery, given that they are such complex and dynamic environments.

IP networks are comprised of devices such as routers, switches, and firewalls that are interconnected by network links. These devices are not "plug-and-play," rather, they must be provided with specific instructions, also known as scripts or configurations, which indicate exactly how they are to interact with each other to provide the correct end-to-end IP network service. This is why we refer to IP device configurations as the DNA of the network—they literally control the network's behavior.

Unfortunately, there is nothing simple or standard about these configurations. Each one must be manually programmed into the network devices, and every vendor uses a different configuration language for its devices. Furthermore, device configurations change virtually everyday in response to new application deployments, organizational or policy changes, new device or technology deployments, device failures, or any number of other reasons. Device configurations have an average of 500 lines of code per device. A Fortune 500 enterprise that relies on an IP can easily have over 50 million lines of configuration code in its network. But numbers of devices and lines of code are only part of the problem. Configurations can contain parameters for about 20 different IP protocols and technologies that need to work together. Those protocols and technologies must satisfy various, constantly changing service requirements, some of which are inherently contradictory, such as security and connectivity with the Internet. So configuration errors can easily occur due to entry mistakes, feature interaction, poor process, or lack of a network-wide perspective.

The labor-intensive and constantly changing nature of IP network operations is analogous to software development. The key difference, as illustrated in Figure 11.1, is that software development has matured to the point where errors are significantly reduced by having different people responsible for requirements, code writing, and testing. More importantly, testing in software development is a well-established process, while there is no similarly rigorous process in IP network deployment and operation. The impacts of configuration errors are well documented. The National Security Agency (NSA) found that 80 percent of the vulnerabilities in the Air Force were due to configuration errors, according to a recent report from Center for Strategic and International Studies (CSIS).[1] British Telecom (BT)/Gartner has estimated that 65 percent of cyber attacks exploit systems with vulnerabilities introduced by configuration errors.[2] The Yankee Group has noted that configuration errors cause 62 percent of network downtime.[3] A 2006 Computer Security Institute/FBI computer crime survey conservatively estimates average annual losses from cyber attacks at $167,000 per organization. [4]

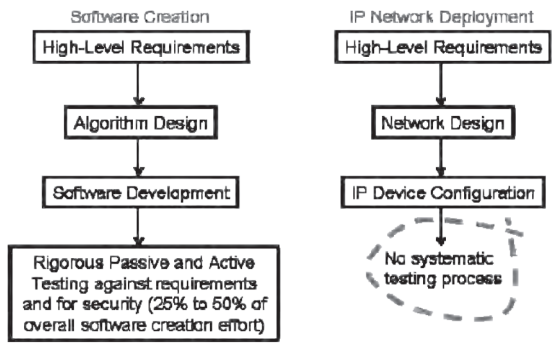


Figure 11.1. Inadequate Testing in IP Network Deployment Compared to Software Development.

CONFIGURATION ERRORS FOUND IN OPERATIONAL IP NETWORKS

IP network configuration errors are hard to detect since they can require validation of multiple protocols and device configurations simultaneously. These errors typically remain latent until they are exploited by cyber attackers, discovered by auditors, or result in network failures. Below are specific examples of configuration errors, and their potential impact on the organization. Many of these errors have been discovered in operational networks while performing configuration assessments.

Reliability.

Organizations that depend on the IP network to provide a very reliable service have to ensure that there are no single points of failure in the network. It is not sufficient to just provide redundant network devices and links at the physical level. It is also critical to ensure that the configurations of the network devices make use of the available redundant physical resources, and that the redundancy is ensured across multiple layers. Examples of misconfigurations that result in single points of failure include:

- Mismatched device interface parameters. This mismatch prevents devices from establishing logical connectivity even though physical connectivity exists.
- Hot Standby Routing Protocol (HSRP) inconsistently configured across two routers that are expected to mirror each other. The standby router will not take over when the main router fails.

- Access Control Lists (ACLs) or firewall rules stop specific application traffic on a path. Even if the path provides redundancy in general, the ACLs/rules still are a cause for a single point of failure to exist for the specific application traffic.
- Use of a single Open Shortest Path First (OSPF) Area Border Router (ABR). The OSPF areas that are connected by the ABR will become isolated if the ABR fails.
- Multiple VPN connections sharing a single physical link or device. The redundancy expected from the multiple VPN connections is not provided due to their dependence on a single physical resource.

In addition to the errors that introduce single points of failures as described above, other errors in configuration of IP routing protocols, such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Multi-Protocol Label Switching (MPLS), and Intermediate System to Intermediate System (IS-IS), can also impact network reliability. Examples of such errors include:

- Inconsistent routing parameters such as OSPF Hello and Dead interval across multiple routers. OSPF will not function efficiently if such parameters are inconsistent, resulting in ephemeral traffic loops and poor network performance.
- Best practices that are proposed by vendors and experts for routing protocols, such as the use of a full-mesh to connect all internal BGP (iBGP) routers, and that OSPF route summarizations should include IP addresses of all interfaces ex-

cept the loopback interface of a router, are not followed. Not adhering to best practices generally results in an unstable network that will have intermittent connectivity issues that are difficult to debug.
- Use of inappropriate IP addresses, such as addresses assigned to other organizations, or private addresses in parts of the network directly exposed to the Internet. Such networks will start advertising routes for IP addresses they do not own, resulting in Internet routing issues.

Security.

The most obvious configuration errors in this category can be found in firewalls, in the form of "holes" that are inadvertently left in firewall configurations. These holes are actually rules that permit specific application traffic to pass temporarily through the firewall, and then these rules are not removed after they are no longer needed. Cyber attackers scanning enterprise networks discover these holes and craft their attacks on the enterprise infrastructure through them. Apart from the obvious firewall holes, there are several other examples of errors that impact security, such as:

- Static route on device does not direct application traffic into IPSec tunnel. This results in sensitive traffic remaining unprotected as it transits the network instead of flowing through the secure IPSec tunnel.
- Best practices for Virtual LAN (VLAN) security, such as disabling dynamic-desire and using root-guard and Bridge Protocol Data Unit (BPDU)-guard on switch access ports, are not

followed. Leaving the dynamic-desire VLAN feature enabled in a switch allows an attacker that connects to the switch to monitor all traffic passing through the switch.

- Link left active between devices. If the devices belong to network segments that are not meant to have a direct connection, then a backdoor has been introduced that can be exploited by attackers.
- Mismatched IPSec end-points. This results in sensitive traffic remaining unprotected as it transits the network instead of flowing through the secure IPSec tunnel.
- Adequate authentication is not used between devices for exchanging routing protocol information. An attacker can connect to a network device and extract or inject spurious routing information.

Quality of Service.

IP traffic with demanding network latency and packet-loss rate requirements, such as Voice over IP (VoIP) and financial services applications, requires appropriate Differentiated Services and other QoS configurations in the network devices. In a large network, it is easy to make errors in the QoS configurations. Examples of such errors include:

- Incorrect bandwidth or queue allocation on device interfaces for higher priority traffic. During high-load periods, higher priority traffic will not receive its due bandwidth or queue, resulting in higher latency or packet-loss.
- Inconsistent QoS policy definitions and usage across multiple devices. The same QoS policy

may be implemented differently across multiple devices, resulting in application traffic receiving different treatment at the different devices, which can impact latency and packet-loss during periods of high-load.

REGULATORS EXPECT COMPLIANCE

The world's growing reliance on IP and the highly networked nature of government computing environments have also motivated a wave of regulations to improve security, reliability, and QoS.

In the United States, the Federal Information Security Management Act (FISMA) of 2002 requires federal agencies to develop, document, and implement security programs.[5] Office of Management and Budget (OMB) Circular A-130 (an implementation guideline for FISMA), establishes, among other things, a minimum set of controls to be included in automated, inter-connected information resources.[6] The National Institute of Standards and Technology (NIST) has promulgated security requirements, for protecting the confidentiality, integrity, and availability of federal information systems and the information handled and transmitted by those systems.[7] NIST's "Guideline on Network Security Testing"[8] recommends that security testing be a routine part of system and network administration. It also directs organizations to verify that systems have been configured based on appropriate security mechanisms and policy. In addition, laws such as the Sarbanes-Oxley Act and Health Insurance Portability and Accountability Act, among others, are fueling the push for network protection.

Outside the United States, organizations such as the British Standards Institute (BSI), International Or-

ganization for Standardization (ISO), and Information Technology Infrastructure Library (ITIL) recognize the complexity of IP networks and the importance of security. In 2006, BSI published "Delivering and Managing Real World Network Security," which explains that networks must be protected against malicious and inadvertent attacks and "meet the business requirements for confidentiality, integrity, availability, non-repudiation, accountability, authenticity and reliability of information and services. [9]

MANY ASSESSMENT APPROACHES PROVE DEFICIENT

Many of the solutions for IP network configuration assessment have proven woefully inadequate. They fall generally into three categories: manual assessment, invasive systems, and noninvasive systems.

Manual Assessments.

In many organizations IP device configurations are large, complex software systems that depend on a hands-on, highly skilled administrator base for creation, update, and troubleshooting. Given the size of many networks and the cost of labor, the manual approach has obvious limitations. One large U.S. federal agency, for example, has 10 five-person teams handling manual analysis of device configurations for its 120 locations.

Invasive Systems.

Invasive scanning solutions, such as ping, traceroute, and their commercial variants, send traffic to

devices in the network and use the responses to assess compliance. Such approaches work for simple usage, such as demonstrating IP connectivity between network nodes and identifying the software version on the devices. However, when it comes to rigorous assessment, they have serious shortcomings, including:

- No root-cause analysis. They can detect problems, but offer little, if any, help in diagnosing the configuration errors that caused them.
- Nonscalable. They cannot deliver "all" or "none" results, which generally require a huge number of tests. For example, to confirm that "There is connectivity between all pairs of internal subnets," connectivity tests are required for the number of subnets squared.
- No testing requirements on contingencies. Contingencies may be security breaches, component or link failures, changes in traffic conditions, or changes in requirements themselves. It is impractical to simulate contingencies on a network that supports real-time and critical services. For example, to detect the existence of a single point of failure, one would have to fail each device and check whether the end-to-end requirement still holds.
- Potential to disrupt network operations. Invasive scanning can introduce malware into a network, or inadvertently exploit a vulnerability that brings down a device.

Noninvasive Systems.

Noninvasive solutions include network simulation tools and Network Change and Configuration Management (NCCM) systems, which are analogous to software version control systems like Concurrent

Versioning System (CVS) and Source Code Control System (SCCS). Such systems tend to treat configurations as "blobs," and support IP device configuration backups, upgrades, controlled rollbacks, and maintainability of device configurations. While these capabilities are important, they are not sufficient for detecting issues that must be proactively resolved to ensure that the network continuously satisfies service requirements.

Noninvasive assessment is preferable to manual and invasive assessment because it does not impact ongoing network operations, but many of the existing systems have limitations, including:

- Individual-device assessments. Configuration management tools assess individual devices in isolation using a template-based approach, even though structural vulnerabilities are often created by interactions between protocols across multiple devices. Even nonsecurity protocols can interact improperly to create structural vulnerabilities. For example, if redundant tunnels traverse the same physical router and that router fails, all tunnels fail.
- Nonscalable. Certain types of requirements, such as reachability, can be assessed by network simulation tools; however they can take hours to compute reachability for networks with more than 50 devices, because they simulate each and every transition of the state machine of each protocol, whether it is for routing, security, reliability, or performance.

TOWARD A SOLUTION THAT WORKS

Building on our long history of involvement in assuring all types of communications networks, Telcordia has spent years researching the issues of IP network reliability and security. Part of this work was funded by the Department of Homeland Security's HSARPA office. That work has yielded important insight into the features and functions that an effective IP network configuration assessment solution must have, and all are capabilities that are achievable today.

Desirable Features of a Solution.

A scalable and effective solution for performing IP network assessments to detect configuration errors needs to possess the following features:

- Automatic and proactive network-wide, multidevice, and multivendor assessments against a comprehensive and updatable knowledge base that considers the network in its entirety and not just at a per-device level. The knowledge base should include rules for best current practices, regulatory compliance, and customer-specific requirements.
- Findings should visualize noncompliant rules and devices down to the "root" cause, eliminating speculation about cause.
- Nonintrusive, detailed, multilevel visualizations for physical connectivity, IP subnets, routing, VLAN, VPN, and MPLS. These visualizations, and the service reachability analysis mentioned below, can be computed using graph theory algorithms on data from the configurations.

- Service reachability analysis that visualizes path and single points of failure between network devices without generating traffic on the network.
- Network change impact analysis using the rules knowledge base, so new or changed configurations can be analyzed to detect errors before deployment to devices.
- Automated reconciliation of configuration and inventory information to identify and eliminate inconsistencies and errors.

Configuration Extraction.

Network and security administrators are generally reluctant to share IP network device configurations because they include sensitive information such as passwords and IP addresses. IP address anonymization and password obfuscation tools are of limited benefit since their usage tends to result in critical information being removed and lost from the configurations. The loss of this information makes the configuration assessments less effective. So for any configuration assessment solution to obtain complete configurations from administrators, it needs to provide assurances that their configurations will be adequately protected.

The most effective approach for acquiring configurations is for the configuration assessment solution to have direct read-only access to the IP network devices for extracting the configurations using device vendor-supported technologies such as secure FTP or remote copy. This direct approach ensures that the most current configuration information is securely retrieved without modification by administrators, and any other device-specific data relevant for validation can also be

retrieved. Another approach is to rely on backups of device configurations from a file-system. Most organizations maintain versions of their IP network device configurations on a file system as backups, to be used to recover a device after its failure or for rolling-back configurations after an unsuccessful configuration change. The configuration assessment solution can acquire these backed-up configurations automatically, either periodically or every time new configurations appear in the backup file-system.

Configuration Adaptors.

Supporting the desirable features identified above requires detailed information from the device configurations. Since every IP network device vendor has their individual configuration language, software adaptors are needed that can extract the detailed information from vendor-specific format (e.g. Cisco IOS, Checkpoint, etc.), and convert the information into a vendor-neutral representation. Based on our experience, the adaptors need to extract as many as 750 attributes from a single configuration to support the desired features, as compared to less than 100 that are extracted by NCCM systems.

Telcordia IP Assure.

Telcordia's IP Assure solution (www.telcordia.com/products/ip-assure) satisfies many of the requirements discussed previously for IP network configuration validation. Figure 11.2 illustrates the high-level information flow that is supported in IP Assure. Solution details can be obtained by contacting the author.

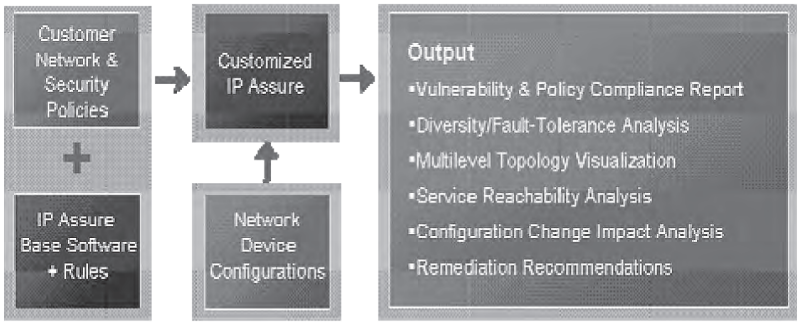


Figure 11.2. Telcordia IP Assure Information Flow.

SUMMARY

IP networks are no longer optional throughout the business and government sectors. This fact, along with the emergence of international regulations on security, reliability, and QoS, means that IP network assessment is a necessity. Many existing solutions on the market, including troubleshooting by skilled administrators, traffic-based vulnerability and penetration testing, NCCM software, and network simulation tools, do not (and cannot) fulfill the world's increasingly rigorous objectives. However, the technology exists today for a nonintrusive and comprehensive IP network assessment solution. Such a solution can provide auditable validation of regulations, eliminate IP network downtime caused by configuration errors, and stop the cyber attacks that exploit those errors. Telcordia IP Assure is an example of such a solution that is available today.

ENDNOTES - CHAPTER 11

1. "Securing Cyber Space for the 44th Presidency," Washington, DC: The Center for Strategic and International Studies (CSIS), December 2008, available from *www.csis.org/media/csis/pubs/081208_securingcyberspace_44.pdf*.

2. British Telecom/Gartner Study, "Security and business continuity solutions from BT," available from *www.btnet.cz/business/global/en/products/docs/28154_219475secur_bro_single.pdf*.

3. *Ibid*.

4. L. Gordon *et al.,* CSI/FBI Computer Crime and Security Survey, 2006, available from *www.cse.msu.edu/~cse429/readings06/FBI2006.pdf*.

5. Federal Information Security Management Act (FISMA) of 2002, available from *csrc.nist.gov/policies/FISMA-final.pdf*.

6. "Security of Federal Automated Information Resources," OMB Circular A-130, Appendix III, available from *www.whitehouse.gov/omb/circulars/a130/a130appendix_iii.html*.

7. "Minimum Security Requirements for Federal Information and Information Systems," FIPS-200, published by NIST, available from *csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf*.

8. "Guideline on Network Security Testing," SP800-42, published by NIST, available from *csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf*.

9. "Delivering and Managing Real World Network Security," British Standards Institute, 2006, available from *www.bsi-global.com/en/Standards-and-Publications/Industry-Sectors/ICT/ICT-standards/BIP-0068/*.

CHAPTER 12

ON THE NEW BREED OF DENIAL OF SERVICE (DOS) ATTACKS IN THE INTERNET

Nirwan Ansari
Amey Shevtekar

INTRODUCTION

Denial of Service (DoS) attacks impose serious threats to the integrity of the Internet. These days, attackers are professionals who are involved in such activities because of financial incentives. They bring higher sophistication to the attack techniques that can evade detection. A shrew attack is an example of such a new threat to the Internet; it was first reported in 2003, and several of these types of attacks have emerged since. These attacks are lethal because they can evade traditional attack detection systems. They possess several traits, such as low average rate and the use of TCP as attack traffic, which empowers them to evade detection. Little progress has been made in mitigating these attacks. This chapter presents an overview of this new breed of DoS attacks along with proposed detection systems for mitigating them. The analysis will hopefully lead to a better understanding of these attacks, and help to stimulate further development of effective algorithms to detect such attacks and to identify new vulnerabilities which may still be dormant.

The Internet has become an integral part of various commercial activities like online banking, online shopping, etc. However, the Internet has been plagued by a variety of security threats over the past several years.

The Distributed Denial of Service (DDoS) attacks received much attention after 2000 when yahoo.com was attacked. After that event, DDoS attacks have been rampaging throughout the Internet. DDoS News,[1] has since been keeping track of DDoS related news. A tremendous amount of work has been done in the industry and academia to mitigate DDoS attacks, but none have been able to successfully eradicate them. The motivations for launching attacks have shifted significantly. Initially, they were for publicity, but now they are for economic or political incentives. Thus, DDoS attacks are a prevalent and complex problem.

Figure 12.1 depicts the trend of DDoS attacks in the Internet. The x-axis indicates the efficiency of the attack, indicating how much damage it can cause to the good traffic. The y-axis indicates the detectability of the attack, indicating the exposure of the attack to defense systems. The early brute force attacks relied on sending high rate attack traffic continuously to a website. They are now easily detected because many defense systems can distinguish such anomalous attack traffic.[2] The shrew and Reduction of Quality of Service (RoQ) attacks are emerging low rate DoS attacks that are difficult to detect as compared to the brute force attack, but they primarily affect only long-lived TCP traffic. One major contribution of this chapter is to forewarn and model the emerging sophisticated attacks, because attacks have a higher impact on good traffic and yet they are very evasive.

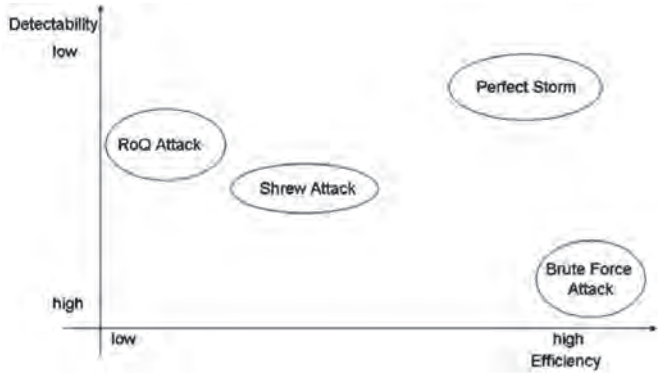


Figure 12.1. The Trend of DDoS Attacks in the Internet.

TRADITIONAL BRUTE FORCE ATTACKS

A Denial of Service (DoS) attack is defined as an attack that prevents a network or a computer from providing service to the legitimate users.[3] It typically targets the bandwidth of the victim. A DDoS attack is defined as an attack that uses multiple unwilling computers to send the attack traffic to the victim. A DDoS attack is more lethal since it exerts a large capacity of attack traffic as compared to a DoS attack. These attacks are also referred to as brute force attacks, since they send attack traffic at high rates and lack characteristics required to be stealthy. There are several types of brute force DDoS attacks that have been reported in the literature, a few of the commonly used attacks are described below. DDoS attacks can be characterized as shown in Figure 12.2.[4] DDoS attacks can be classified by the degree of automation, i.e., the level of sophistication of the attack mechanism.

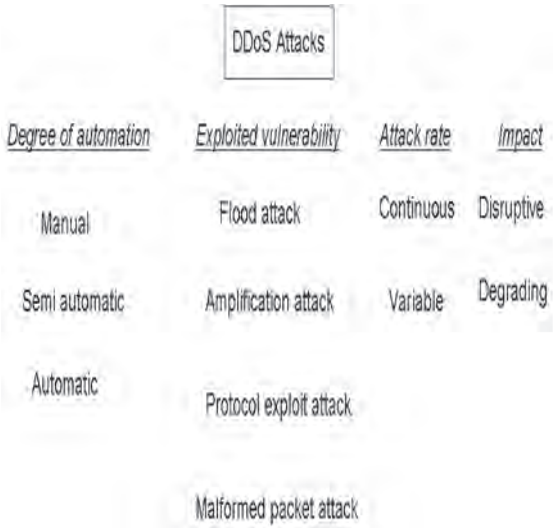


Figure 12.2. Classification of DDoS Attacks.

Early attacks were manual, and were improved gradually. In a manual attack,[5] the victims are scanned for a particular vulnerability which is exploited by the attacker to gain access into the victim's system. An attacker then use commands to control the victim during the attack. In a semi-automatic attack, some steps of the attack procedure, which were originally manually performed become automated; for example, some of the victims are compromised to act as attack agents who coordinate the attack by issuing commands to conceal the identity of the real attacker even if the attack is detected.[6] The attack agents are preprogrammed with the necessary required commands. All of the recent attacks have been highly automatic requiring minimal communication between the attacker and the compromised machines once the attack was launched. All the attack steps are preprogrammed and delivered as a payload to infect clients, also referred to as zombies or

bots. Some new attack payloads, which fail to detect a specific vulnerability in a victim machine, will automatically scan for another vulnerability in the same machine. Recent botnet attacks on Estonia's websites employed fully automated mechanisms.[7] Botnet is a network of bots or zombies controlled by a botmaster.[8]

Classification of DDoS attacks based on an exploited vulnerability takes into account the property of the network or the protocol used in the attack. The category to which the flood attack belongs is the simplest of all categories and is one in which an attacker relies on denying the network bandwidth to the legitimate users. The common example of this category is the UDP flood attack.[9] In a UDP flood, an attacker sends UDP packets at a high rate to the victim so that the network bandwidth is exhausted. UDP is a connectionless protocol, and therefore it is easy to send UDP packets at any rate in the network. Another attack in this category is the ICMP echo flood; it involves sending many ICMP echo request packets to a host. The host replies with an ICMP echo reply to each of the two ICMP echo request packets, and many such requests and reply packets fill up the network bandwidth.

In the category of amplification attack, an attacker exploits a protocol property such that few packets will lead to amplified attack traffic. The DDoS "SMURF" attack, which exploits the ICMP protocol,[10] falls in this category. It involves replacing a source IP address of the ICMP echo request packet with the address of the victim. The destination address of the ICMP echo request is the broadcast address of the LAN or so-called directed broadcast addresses. On receiving such a packet, each active host on a LAN responds with an ICMP echo reply packet to the victim. Typically, a LAN has many active hosts, and so a tremendous amount

of attack traffic is generated to cripple the victim. To avoid such an attack, most system administrators are advised to disable the directed broadcast addresses.

In the category of protocol exploit, a property of the protocol is exploited. The SYN attack[11] exploits the TCP protocol's three-way handshake mechanism. Web servers use port 80 to accept incoming HTTP traffic that runs on top of the TCP protocol. When a user wants to access a webpage, it sends a SYN packet to the web server's open port 80. The web server does not know the user's IP address before the arrival of the SYN packet. The server, upon receipt of a SYN packet, sends a SYN/ACK packet, and thus puts the connection in the LISTEN state. A legitimate user's machine replies to the web server's SYN/ACK packet with an ACK packet and establishes the connection. However, if the SYN packet has been sent from an attack machine which does not respond to the server with an ACK packet, the web server never gets an ACK packet and the connection remains incomplete. Every web server has a finite amount of memory resources to handle such incomplete connections. The main goal of the SYN attack is to exhaust the finite amount of memory resources of a web server by sending a large number of SYN packets. Such an attack causes the web server to crash. Another similar protocol exploit attack is the PUSH + ACK attack,[12] which also falls in this category.

In the category of malformed packet attack, the packet header fields are modified to instigate a crash of the operating system of the receiver. An IP packet having the same source and destination IP address is a malformed packet.[13] In another kind of malformed packet attack, the IP options fields of the IP header are randomized, and the type of service bit is set to one. A ping of death attack involves sending a ping

packet larger than the maximum IP packet size of 65535 bytes.[14] Historically, a ping packet has a size of 56 bytes, and most of the systems cannot handle ping packets of a larger size. Operating systems take more time to process such unusual packets, and a large quantity of such packets can crash the systems.

DDoS attacks can also be classified by their attack rates, namely: continuous vs. variable. Likewise, they can also be classified by their impacts: disruptive vs. degrading. Disruptive attacks aim for denial of service while degrading attacks aim for reduction of quality.

NEW BREED OF STEALTHY DoS ATTACKS

Internet security is increasingly more challenging as more professionals are getting into this lucrative business. An article in the *New York Times*,[15] describes one such business of selling the software exploits. Attacks are also getting more sophisticated, as the attackers are not merely interested in achieving publicity. The shrew attack is one such intelligent attack, which was first reported in Low-Rate TCP-Targeted Denial of Service Attacks in "The Shrew vs. the Mice and Elephants,"[16] followed by a series of variants.[17] This study considers these attacks as low rate DoS attacks. It is typically illustrated by a periodic waveform shown in Figure 12.3, where T is the time period, t is the burst period, and R is the burst rate.

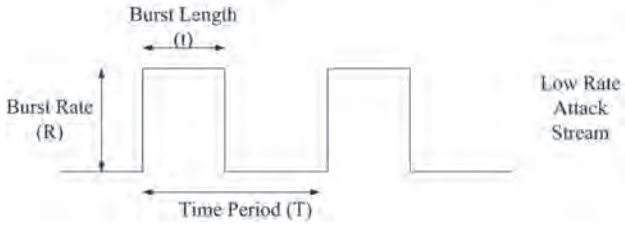


Figure 12.3. An Example of a Generic Low Rate DoS Attack Pattern.

A shrew attack exploits widely implemented minimum RTO,[18] property of the TCP protocol. The following characterize the low rate TCP DoS attack:

- It sends periodic bursts of packets at one-second intervals.
- The burst rate is equal to or greater than the bottleneck capacity.
- The burst period is tuned to be equal to the round-trip times of the TCP connections; this parameter determines whether the attack will cause DoS to the TCP connections with small or long round-trip times.
- The exponential back off algorithm of the TCP's retransmission mechanism is eventually exploited.

In a Reduction of Quality of Service (RoQ) attack,[19] the attacker sends high rate short bursts of the attack traffic at random time periods, thereby forcing the adaptive TCP traffic to back off due to the temporary congestion caused by the attack bursts. In particular, the periodicity is not well defined in a RoQ attack, thus allowing the attacker to keep the average rate of the attack traffic low in order to evade the regulation of adaptive queue management like RED and

RED-PD.[20] By sending the attack traffic, the RoQ attack introduces transients and restricts the router queue from reaching the steady state. The awareness of these stealthy attacks demands early fixes. For simplicity, the term "low rate DoS attack" refers to both the shrew and the RoQ attack, unless otherwise stated as shown in Figure 12.3. The attacker can also use different types of IP address spoofing to evade several other detection systems. Owing to the open nature of the Internet, IP address spoofing can still evade ingress and egress filtering techniques at many sites.[21] A low rate DoS attack can use IP address spoofing in a variety of ways like random IP address spoofing and continuous IP address spoofing.[22] The use of IP address spoofing most importantly divides the high rate of a single flow during the burst period of the attack among multiple flows with spoofed identities. This way, an attacker can evade detection systems that concentrate on finding anomalous traffic rate. The detection systems that rely on identifying periodicity of the low rate DoS attack in the frequency domain can detect the periodicity, but they fail to filter the attack traffic because it is difficult to know the IP addresses that an attacker will use in the future.

This problem is further exacerbated by the use of botnets; a botnet is a network of compromised real hosts across the Internet controlled by a master.[23] Since an attacker using botnets has control over thousands of hosts, it can easily use these hosts to launch a low rate DoS attack; this is analogous to a low rate DoS attack that uses random or continuous IP address spoofing. Now, with the use of botnets, the IP addresses of bots are not spoofed and so these packets cannot be filtered by spoofing-prevention techniques. In fact, these attack packets are similar to the HTTP

flows. This random and continuous IP address spoofing problem described above is unique to the low rate DoS attacks, and is different from other types of DDoS attacks. These attacks[24] can be launched from any routers in the Internet; the edge routers can be easy targets as their capacities are small, and hence attackers can easily incite denial of service to the VoIP users traversing those routers. Low rate DoS attacks fall in the DDoS attack category of variable attack rate and degrading impact.

The perfect attack[25] is the latest attack model which is extremely lethal as compared to the attack models discussed before. The perfect attack has the ability to disguise itself as a normal traffic, thereby making detection difficult. It relies on using readily available botnets to send the attack traffic. Botnets are formed at an alarming rate today because of increasing vulnerabilities in various software applications. The users of these applications are often average users who are not security conscious, thus leaving their systems exposed to exploitations. Social engineering attacks are also used to increase the bot population. Thus, all these conditions create a breeding ground for rogues to develop new attacks. The perfect attack model consists of two parts: an initial, short, deterministic high-rate pulse, and a feedback-driven sustained attack period with a network-adaptive attack rate, as depicted in Figure 12.4.

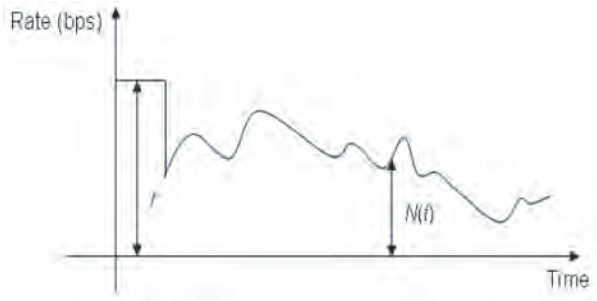


Figure 12.4. Attack Traffic for a Perfect DDoS Attack.

To accomplish the objectives identified above, a perfect attack is envisioned to be executed as follows. The attack traffic is injected from a botnet toward a target with a bottleneck queue as described in Figure 12.4. Initially, a high-rate pulse is sent at a rate of r for a duration of τ that overflows the buffer such that all packets are dropped. This pulse is similar to a shrew pulse, with the main difference that it only occurs at the beginning, once or twice, to drop all the packets in the queue. Thus, after this pulse, it is assumed that all long-lived TCP flows are in a timeout state, and thus do not send traffic. The only legitimate traffic that arrives immediately after the pulse is flow that is being established.

Thus, in the phase after the pulse, the attacker must fill the bottleneck queue with its own traffic as fast as possible and to sustain this level. Ideally, this filling ensures that only a small fraction of legitimate packets ever passes the bottleneck. Such a high drop rate per flow implies that: (1) a large number of SYN packets are dropped; (2) TCP flows experience packet loss already in slow start; and (3) other non-TCP traffic incurs significant packet loss. Thus, in this second

phase, the attack traffic is sent according to the following pattern. Denote $B(t)$ as the available bandwidth at the bottleneck and C as the link capacity of the attack target link. Then, the sustained attack traffic $N(t)$ Equation (1) is: $N(t) = C + B(t)$

Under the assumption that all or the vast majority of the bottleneck bandwidth is consumed by the attack traffic, Equation (1) aims at maintaining a steady consumption. This attack traffic is the TCP traffic at a rate equivalent to the link capacity C. To fill the bottleneck and to compensate for drops in the TCP attack rate, UDP traffic at a rate of $B(t)$ is injected into the network. Note here that the UDP traffic is a function of the available bandwidth rather than the capacity as in a shrew attack or an RoQ attack. $N(t)$ is periodically updated and adjusted with the time period T. The update contains a rate adaptation but also the chance to exchange the zombies in the attack to create a diverse traffic pattern from different traffic sources. Thus, at the bottleneck, $N(t)$ creates a traffic pattern consisting of a superposition of many TCP flows, with a small fraction of UDP traffic, whose sources vary over time. After each period T, a new set of TCP flows are directed at the bottleneck link. These TCP flows begin in the slow start phase of TCP and end in the slow start phase as well. It is important to keep the attack TCP flows in slow start because they have been shown to affect long-lived TCP flows on shorter timescales, and also introducing a new TCP connection allows the congestion window to grow rapidly, otherwise attack TCP flows will enter congestion avoidance phase and will try to share the bandwidth with the legitimate TCP flows. The period T can be random so as to evade detection particularly for systems that try to find the deterministic attack pattern. Note that this interplay

between TCP and UDP further complicates the detection. In contrast to the shrew attack where the repetitive pulses can be relatively detected, the perfect attack does not create such a repetitive pattern. Instead, the dynamics lead to an ever changing traffic pattern that cannot be observed and captured by the defense.

DEFENSE SYSTEMS

Mitigating DDoS attacks is a widely studied problem, and some of the popular approaches are described below. Defense systems can be broadly classified based on their functions. There are four main categories of defense systems: intrusion prevention, intrusion detection, intrusion response, and intrusion mitigation.[26]

Intrusion prevention systems prevent an attack from occurring. Ingress and egress filtering control IP address spoofing that is used in the attack. Ingress filtering only allows packets destined for the source network to enter the network, thereby filtering all other packets. It is implemented at the edge routers of the network, and it limits the attack traffic from entering the network. Egress filtering is an outbound filter that allows only packets with source IP addresses originated from the source network to exit the source network. Use of egress filtering controls attack traffic going to destination networks.[27] Disabling IP broadcasts prevents smurf attacks. Honeypots are network decoys,[28] that study attack behavior before the onset of an attack. Honeypots act as early warning systems. Honeypots mimic all aspects of a real network like a web server and a mail server to lure attackers. The primary goal of the honeypots is to determine/derive the exploit mechanism of the attack in order to build

defense signatures against the exploit. Intrusion prevention systems cannot completely prevent an attack, but they contain the damage of the attack. They allow building better defense systems by analyzing the attack.

Intrusion detection systems detect an attack based on attack signatures or anomalous behaviors. Snort is a popular signature based network intrusion detection system.[29] It performs protocol analysis and content matching to passively detect a variety of attacks like buffer overflows, port scans, and web application attacks. Snort uses Berkeley's libpcap library to sniff packets. It uses a chain structure to maintain rules. The header of each chain is a tuple of source IP address, destination IP address, source port, and destination port. Various rules are then attached to the header so that packet information is matched to a header and the corresponding rules to detect an intrusion.

Anomaly detection systems rely on detecting shift in the normal traffic patterns of the network. The Reference A network management system is widely deployed in the Internet and is effectively used for intrusion detection. Consider the ping flood attack in which many ICMP echo request packets are sent to the target. The SNMP ICMP MIB group has a variable icmpInEchos, which shows the sudden increase in its count during the ping flood attack. During the UDP flood attack, SNMP UDP MIB group's udpInDatagrams shows a similar increase in its count. To detect localized variations in important MIB variables, a time series is segmented in small sub-time series which are compared to the normal profiles. In a DDoS attack, variations are so intense that averaging the time series on properly chosen time intervals enables anomaly detection.

The examples discussed above are network based intrusion detection systems, but intrusion detection can also be performed at a host. A data mining based approach is one such method.[30] Datasets consisting of normal and abnormal data points are gathered and fed to a classification algorithm to obtain classifiers. Once trained on these classifiers the training datasets are then used to find abnormal data points. D-WARD[31] is an intrusion detection system to be installed at the network edges to detect attack sources. It monitors network traffic rates to determine asymmetry in the traffic rate. Typically in a DDoS attack like the SYN attack, there are more SYN packets leaving the network as compared to ACK packets entering the network. D-WARD attempts to stop the attacks close to the sources so that network congestion is reduced. Attack traffic even affects traffic not intended for the victim, and thus D-WARD also minimizes the collateral damage from the attack. Figure 12.5 shows the conceptual diagram of the PPM scheme where each router marks packets probabilistically so that the victim can reconstruct the entire path from the source router.[32]

Intrusion response systems are required to find the source of the attack in order to stop the attack. Blocking the attack traffic is sometimes done manually by contacting network administrators who change the filtering policies to drop the attack traffic at routers. If an attacker is using source IP address spoofing, manual filtering is not useful and schemes like IP traceback are required. IP traceback traces the IP packets back to their sources and helps reveal attack sources.[33] In probabilistic packet marking (PPM)[34] shown in Figure 12.5, routers mark their addresses on packets that traverse through them. Packets are selected randomly with some fixed probability of marking. Upon receiving many packets a vic-

tim can construct the route back to the sources by reading router marks. Router vendors need to enable the marking mechanism, so ISP participation is required. This scheme does not require additional overhead bandwidth, which is an important advantage of this scheme. In contrast, a scheme referred to as deterministic packet marking (DPM),[35] shown in Figure 12.6, only marks packets passing through edge routers of the network. At the victim, a table is maintained for mapping between source addresses and router interface addresses. This facilitates the reconstruction and identification of the source of the packets. Some countermeasures to mitigate the low rate DoS attacks in the Internet have been reported although none of them has made a comprehensive attempt to address such attacks with IP address spoofing.

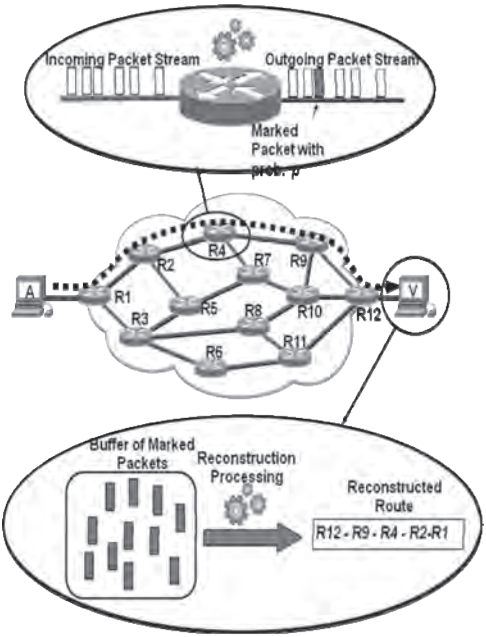


Figure 12.5. Conceptual Diagram of the PPM Scheme.

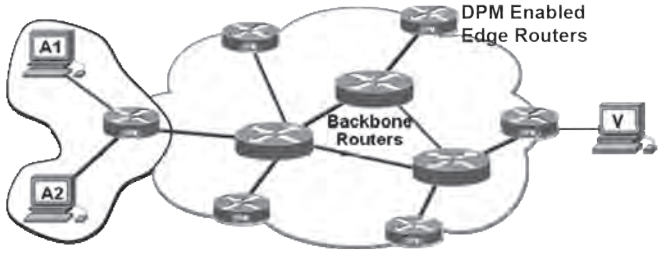


Figure 12.6. Conceptual Diagram of the DPM Scheme.

Pushback, also known as aggregate congestion control scheme (ACC),[36] drops DDoS attack traffic by detecting the attack and sends signals to drop the attack traffic closer to the source as shown in Figure 12.7. The rationale behind the pushback scheme is that the attack traffic has a unique signature in an attack, where the signature consists of identifiers such as port numbers, IP addresses, and IP prefixes. By detecting a signature in the aggregate attack traffic, upstream routers can be instructed to rate-limit flows that match the signature. A router in a pushback scheme has two components: a local ACC mechanism and a pushback mechanism.

The local ACC mechanism is invoked if the packet loss percentage exceeds a threshold of 10 percent. It then tries to determine the aggregate congestion signature and correspondingly tries to rate limit the aggregate traffic. If the rate limiting does not reduce the arrival rate of the attack traffic below a predefined target rate, ACC invokes the pushback mechanism which sends pushback messages to the upstream routers to filter the attack traffic. By repeating this scheme upstream, pushback aims at rate-limiting the attack traffic at the source network. Throttling is another approach to defend web servers from a DDoS

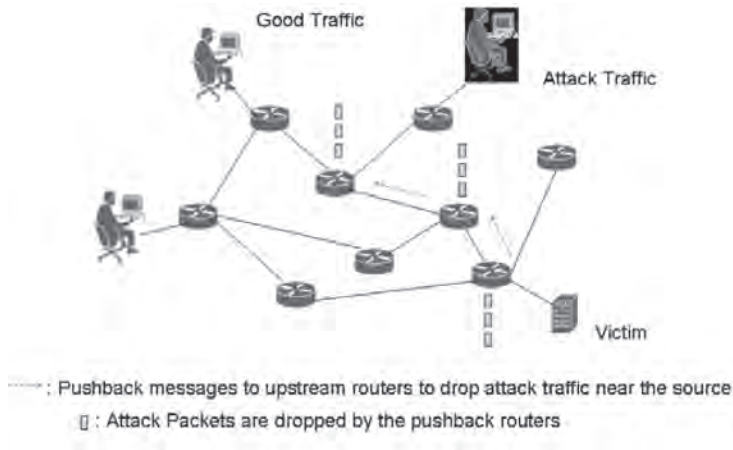


Figure 12.7. Conceptual Diagram of the Pushback Scheme.

attack. It uses max-min fairness algorithm to compute the rate to drop excess traffic. Upstream routers also participate in the scheme so as to drop the attack traffic near the source. There have been several ways to mitigate specific DDoS attacks like SYN attacks. A technique, referred to as a SYN cookie, avoids giving server resources to the SYN packet until a SYN/ACK is received.[37]

The autocorrelation and dynamic time warping algorithm[38] relies on the periodic property of the attack traffic to detect the low rate DoS attacks. It proposes a deficit round-robin algorithm to filter the attack flows; however, it fails to drop attack packets when the attacker uses the continuous cycle and randomized IP address spoofing since each attack flow is a combination of multiple flows and each will be treated as a new flow. Thus, the attacker can easily evade the filtering mechanism. The randomization of RTO proposed to mitigate the low rate TCP DoS attack cannot

defend against the RoQ attack, which targets the network element rather than the end host. The main idea reported in "Collaborative Detection and Filtering of Shrew DDoS Attacks using Spectral Analysis," *Journal of Parallel and Distributed Computing,*[39] is to randomize the minimum RTO instead of setting it to be one second. However, it ignores the advantages of having the minimum RTO of one second, which was chosen as a balance between an aggressive value and a conservative value.

The Collaborative Detection and Filtering scheme proposed in, "Collaborative Detection and Filtering of Shrew DDoS Attacks using Spectral Analysis,"[40] involves cooperation among routers to throttle and push the attack traffic toward the source. They rely on the autocorrelation property to distinguish the periodic behavior of attack traffic from legitimate traffic. Thus, it needs extra DSP hardware for implementation and extra memory to store the flow information of the attack packets to be dropped. The scheme maintains a malicious flow table and a suspicious flow table, which can be overwhelmed under the presence of the IP address spoofing. The novel part is the cumulative traffic spectrum that can distinguish traffic with and without the attack. In the traffic spectrum with the attack, the energy is found more localized at lower frequencies. The attacker can randomize the attack parameters in the RoQ attack. This work does not provide clear guidelines to activate the filtering of attack packets.

The wavelet based approach identifies the abnormal change in the incoming traffic rate and the outgoing acknowledgments to detect the presence of low rate TCP DoS attacks.[41] This approach cannot regulate the buffer size so that the attack flows can be detected as high rate flows by the RED-PD filter, and therefore

the approach was subsequently dropped.[42] This work does not consider the RoQ attack in their analysis; it is difficult for this approach to detect the RoQ attack because the average rate of an RoQ attack is very low. The buffer sizing scheme fails if an attacker uses IP address spoofing because the high rate attack flow is a combination of multiple low rate individual flows. A modified AQM scheme referred to as HAWK[43] works by identifying bursty flows on short timescales, but lacks good filtering mechanisms to block the attack flows that can use the IP address spoofing. This approach can penalize the legitimate short bursty flows, thereby reducing their throughput. A filtering scheme similar to HAWK is proposed to estimate the bursty flows on shorter and longer time scales.[44] The main idea is to use per-TCP flow rate as the normal rate, and anything above that rate is considered abnormal. The identification of flow rates is done online. On a shorter time scale, it is very easy to penalize a normal flow as a bursty flow. The proposal did not consider the random IP address spoofing, where every packet may have a new flow ID. It uses a very complex filtering technique. With the use of the IP address spoofing, it is difficult to come up with the notion of a flow, because the number of packets per flow can be randomized in any fashion during every ON period.

An edge router based detection system is proposed to detect low rate DoS attacks based on time domain technique.[45] Each edge router acts as an entry and exit point for traffic originating from that local area network; essentially all incoming and outgoing traffic will pass through this point. The proposed detection system can be deployed at the edge routers of a local area network in which the server is present. For illustrative purposes, it is assumed that all clients are outside the local area network in which the server is

present, so that the detection system can monitor all flows connecting to the server.

Figure 12.8 shows the basic layout of the system. It has three basic blocks, namely, flow classifier, object module, and filter. Each block functions as follows. The flow classifier module classifies packets based on the flow ID by means of a combination of the IP source address, IP source port, IP destination address, and IP destination port. Flow information is obtained from these packets, and packets are forwarded as usual by the routing mechanism resulting in no additional delay apart from the lookup delay. The object module consists of various objects for each flow that is monitored. A flow is monitored until it is considered normal. The filter is used to block flows that are identified as malicious by the object module. Consider a DoS attack which tries to exploit protocol shortcomings. The object module maintains per flow information by creating objects per flow called flow objects. A thin data structure layer is designed to keep track of these flow objects. It maintains only those parameters which are exploited in the attack. This thin structure keeps track of information about flows classified as malicious. This information is then relayed to the filter module.

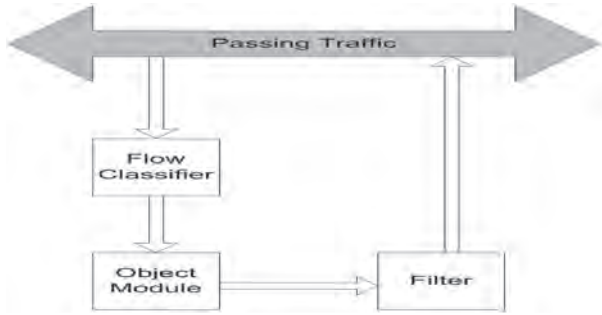


Figure 12.8. The Detection System Architecture.

The flow objects maintain the arrival times of packets at the edge router in the pseudo transport layer. The malicious flow detection submodule of the object module computes the time difference of consecutive packets of each flow. The submodule computes the average high and low of the time difference values. The average high value of the time difference repeats periodically for the attack flow; other flows do not exhibit this property. The malicious flow detection submodule then estimates the burst length of a flow based on the packet arrival times. A flow exhibiting periodicity in the time difference graph is marked malicious, since no legitimate flows will show such periodicity. The time difference technique uses a per-flow approach to store the arrival times of the packets belonging to each flow, and computes the interarrival times between the consecutive packets to detect periodicity. The attacker using IP address spoofing can easily deceive this simple per-flow approach as the time difference approach will not be able to detect periodicity in the attack flow, which is no longer a single flow.

FUTURE RESEARCH DIRECTIONS

Mitigating perfect attack and low rate DoS attacks is extremely critical. Router based solutions that can identify and drop malicious attack traffic can be a possible defense approach. Currently, both perfect and low rate DoS attacks are facilitated by botnets. Mitigation of botnets can be another important step to prevent stealthy DoS attacks. Botnet detection and mitigation is a serious challenge, because attackers find new vulnerabilities at a rapid pace. Secure software development that would be void of vulnerabilities is desirable. These research goals are known and emerging everyday. On the other hand, isolating bots from

accessing the network by using better CAPTCHAs can be another approach to defend against botnets until such a time that we can completely eliminate bots.

CONCLUSION

In this chapter, we have presented a survey of several traditional brute force DoS attacks and examples of more recent stealthy DoS attacks. We have also introduced defense systems discussed in the literature, and identified some of their shortcomings. The focus of this chapter was to present some of the latest advances in the area of DoS attacks to stimulate research for better defense systems and to reveal vulnerabilities that may exist.

ENDNOTES - CHAPTER 12

1. DDoS News, available from *staff.washington.edu/dittrich/misc/ddos/.*

2. C. Douligeris and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art," *Computer Networks*, Vol. 44, No. 5, 2004, pp. 643-666.

3. *Ibid*.

4. *Ibid*.

5. *Ibid*.

6. *Ibid*.

7. DDoS News.

8. D. Dagon, Z. Zhou, W. Lee, "Modeling Botnet Propagation Using Time Zones," Network and Distributed System Security (NDSS) Symposium, 2006.

9. "UDP Port Denial-of-Service Attack," available from *www.cert.org/advisories/CA-1996-01.html.*

10. "Smurf Attack," available from *www.nordu.net/articles/smurf.html.*

11. W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," IETF RFC 4987, 2007.

12. "Push + Ack Attack," available from *www.csie.ncu.edu.tw/~cs102085/DDoS/protocolexploit/push%2Back/description.htm.*

13. Douligeris and Mitrokotsa.

14. "Ping of Death Attack," available from *insecure.org/sploits/ping-o-death.html*.

15. "A Lively Market, Legal and Not, for Software Bugs," available from *www.nytimes.com/2007/01/30/technology/30bugs.html?ex=1327813200&en=99b346611df0a278&ei=5088&partner=rssnyt&emc=rss*.

16. A. Kuzmanovic and E. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants)," ACM SIGCOMM, 2003, pp. 75-86.

17. M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the Transients of Adaptation for RoQ Attacks on Internet Resources," IEEE ICNP, 2004, pp. 184-195; S. Ebrahimi-Taghizadeh, A. Helmy, and S. Gupta, "TCP vs. TCP: a Systematic Study of Adverse Impact of Short-lived TCP Flows on Long-lived TCP Flows," IEEE INFOCOM, 2005, pp. 926-937; X. Luo and R. K. C. Chang, "On a New Class of Pulsing Denial-of-Service Attacks and the Defense," NDSS, 2005; A. Shevtekar and N. Ansari, "Do Low Rate DoS Attacks Affect QoS Sensitive VoIP Traffic?" IEEE ICC, 2006, pp. 2153-2158; R. Chertov, S. Fahmy, and N. Shroff, "Emulation versus Simulation: A Case Study of TCP-Targeted Denial of Service Attacks," Tridentcom, 2006, pp. 316-325.

18. V. Paxon and M. Allman, "Computing TCP's Retransmission Timer," IETF RFC 2988, 2000.

19. Guirguis, Bestavros, and Matta.

20. R. Mahajan, S. Floyd, and D. Wetherall, "Controlling High-Bandwidth Flows at the Congested Router," IEEE ICNP, 2001, pp. 192-201; Y. Xu and R. Guerin, "On the Robustness of Router-based Denial-of-Service (DoS) Defense Systems," *ACM Computer Communications Review*, Vol. 2, 2005, pp. 47-60; S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM, *Transactions on Networking*, Vol. 1, No. 4, 1993, pp. 397-413.

21. R. Beverly and S. Bauer, "The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet," USENIX SRUTI, 2005, pp. 53-59.

22. Xu and Guerin.

23. Dagon, Zhou, and Lee.

24. Guirguis, Bestavros, and Matta.

25. A. Shevtekar, N. Ansari, and R. Karrer , "Towards the Perfect DDoS Attack: The Perfect Storm," IEEE Sarnoff Symposium, 2009, pp. 1-5.

26. Douligeris and Mitrokotsa.

27. P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Address Spoofing," IETF RFC 2827, 2001.

28. W. R. Cheswick, "An Evening with Berferd, in Which A Cracker Is Lured, Endured, And Studied," USENIX Winter Conference, 1992, pp. 163-174.

29. Snort IDS, available from *www.snort.org*.

30. W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," USENIX Security Symposium, 1998, pp. 79-93.

31. J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at Source," IEEE ICNP, 2002, pp. 312-321.

32. A. Belenky and N. Ansari, "On IP Traceback," *IEEE Communications Magazine,* Vol. 41, No. 7, 2003, pp. 142-153.

33. *Ibid*.; Z. Gao, and N. Ansari, "Tracing Cyber Attacks from Practical Perspective," *IEEE Communications Magazine*, Vol. 43, No. 5, 2005, pp. 123-131.

34. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," IEEE/ACM, *Transactions on Networking*, Vol. 9, No. 3, 2001, pp. 226-237.

35. A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Communication Letters*, Vol. 7, No. 4, 2003, pp. 162-164.

36. R. Mahajan, S. Bellovin, S. Floyd, J. Ionnadis, V. Paxson, and S. Shenker, "Controlling High-bandwidth Aggregates in the Network," ACM SIGCOMM CCR, Vol. 32, No. 3, 2002, pp. 62-73.

37. SYN cookies, available from *cr.yp.to/syncookies.html.*

38. H. Sun, J. C. S. Lui, and D. K. Y. Yau, "Defending Against Low-rate TCP Attack: Dynamic Detection and Protection," IEEE ICNP, 2004, pp. 196-205.

39. Y. Chen, K. Hwang, and Y. Kwok, "Collaborative Detection and Filtering of Shrew DDoS Attacks using Spectral Analysis," *Journal of Parallel and Distributed Computing*, 2006, available from *gridsec.usc.edu/files/publications/JPDC-Chen-2006.pdf.*

40. *Ibid*.

41. X. Luo and R. K. C. Chang, "On a New Class of Pulsing Denial-of-Service Attacks and the Defense," NDSS, 2005.

42. S. Sarat and A. Terzis, "On the Effect of Router Buffer Sizes on Low-Rate Denial of Service Attacks," IEEE ICCCN, 2005, pp. 281-286.

43. Y. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: Halting Anomaly with Weighted ChoKing to Rescue Well-Behaved TCP Sessions from Shrew DoS Attacks," ICCNMC, 2005, pp. 2-4.

44. Y. Xu and R. Guerin, "A Double Horizon Defense for Robust Regulation of Malicious Traffic," SecureComm, 2006, pp. 1-11.

45. A. Shevtekar, K. Anantharam, and N. Ansari, "Low Rate TCP Denial-of-Service Attack Detection at Edge Routers," *IEEE Communication Letters*, Vol. 9, No. 4, 2005, pp. 363-365.