ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Defining Threats and its Defensive Methods in Network

Rashmi Pandey*
M.Tech Scholar
Ambedkar Institute of Technology
New Delhi, India
rashmipait@gmail.com

Suresh Kumar
Assistant Professor, Department of CSE
Ambedkar Institute of Technology
New Delhi, India
sureshpoonia@yahoo.com

***Abstract*:** Any user or client under attack in network exhibits jarring deeds called the malicious behavior. In this situation, the entire operation of a network and communication between two clients gets anxious and to exclude such malevolent behavior several security solutions have been discovered. In this paper, threats categories is defined and to defend such behavior, security solutions are offered which are used in furnishing a secure and reliable communiqué in Network.
General Terms: Security, Algorithm

Keywords: Authentication, Authorization, Security, Cryptography, Attacks

I. INTRODUCTION

Security is essentially about protecting assets. Assets may be physical items, such as a Web page or your customer database — or they may be less physical, such as your company's reputation, company's person reputation. Security is a form of protection where a separation is created between the assets and the threat" [1].

A. The Foundations of Security

Security relies on the following elements:

a. Authentication –

It is the process of uniquely identifying the clients or customer of your applications and services. Validation of a user, a computer, or some digital object to ensure that it is what it claims to be [2][4].

b. Authorization –

It is the process that governs the possessions and operations that only authenticated client is permitted to access. Possessions include files, databases, tables, rows, and so on.

c. Auditing –

Effective auditing and cataloguing is the means to non-repudiation. Non-repudiation guarantees that a user cannot deny in future for performing an operation or initiating an operation.

d. Confidentiality –

Confidentiality, also referred to as *solitude or privacy*, is the process of making sure that data remains private and confidential, and that it cannot be disclosed by unauthorized users or eavesdroppers who monitor the traffic flow across a network. Encryption, Access control lists (ACLs) is frequently used to enforce confidentiality.

e. Integrity –

Integrity is the pledge that data is secluded from fortuitous or premeditated modification. Like privacy, integrity is a key concern.

f. Availability –

From a security perception, availability means that Systems remain available for genuine users. The goal for many attackers with denial of service attacks is to hurtle an application or to make sure that it is adequately beleaguered so that other users cannot admittance the application.

II. UNDERSTANDING THREAT CATEGORIES

While there are many variations of specific attacks and attack techniques, it is useful to think about threats in terms of what the attacker is trying to achieve. This changes your focus from the identification of every specific attack — which is really just a means to an end — to focusing on the end results of possible attacks.

A. Stride

Threats faced by the application can be categorized based on the goals and purposes of the attacks. A working knowledge of these categories of threats can help you organize a security strategy so that you have planned responses to threats. **STRIDE** is a system developed by Microsoft for classifying computer security threats. It provides a mnemonic for security threats in six categories [3][5].

a. STRIDE Threats and Countermeasures

Each threat category described by STRIDE has a corresponding set of countermeasure techniques that should be used to reduce risk. These are summarized in Table.1 The appropriate countermeasure depends upon the specific attack [6][9][21].

Table.I STRIDE Threats and Countermeasures

Threat	Countermeasures
Spoofing user identity(attackers pretend to be someone (or something) else)	Use strong authentication. Do not store secrets in plaintext. Do not pass credentials in plaintext over the wire. Protect authentication cookies with Secure Sockets Layer (SSL).

Tampering with data(attackers change data in transit or at rest)	Use data hashing and signing. Use encryption technique. Use digital signatures. Use strong authorization. Use tamper-resistant protocols across communication links. Secure communication links with protocols that provide message integrity.
Repudiation(attackers perform actions that can't be traced back to them)	Create secure audit trails. Use digital signatures.
Information disclosure(attackers steal data in transit or at rest)	Use strong authorization. Use strong encryption. Secure communication links with protocols that provide message confidentiality. Do not store secrets in plaintext.
Denial of service(attackers interrupt a system's legitimate operation)	Use resource and bandwidth throttling techniques. Validate and filter input.
Elevation of privilege(privilege—attackers perform actions they aren't authorized to perform)	Follow the principle of least privilege and use least privileged service accounts to run processes and access resources.

B. Authentication

Depending on your necessities, there are several obtainable authentication mechanisms [7]. If they are not correctly chosen and implemented, the authentication mechanism can expose vulnerabilities that attackers can exploit to gain access to your system. The top threats that exploit authentication vulnerabilities include [7][8][10][12]:

a. Network Eavesdropping

If authenticated permit are passed in plaintext form from client to server, an attacker fortified with elementary network monitoring software on a host on the same network can imprison traffic and obtain user names and passwords.

Countermeasures to prevent network eavesdropping include:

a) Make use of authentication mechanisms that do not broadcast the password over the network such as Kerberos protocol or Windows authentication.
b) Make persuaded passwords are encrypted or use an encrypted communication conduit, for example with SSL.

b. Brute Force Attacks

A brute-force authentication attack is a method of obtaining a user's authentication credentials by guessing usernames and passwords [11].

Brute force attacks rely on computational power to fissure hashed passwords or other secrets secured with hashing and encryption. To mitigate the risk, use strong passwords.

c. Dictionary Attacks

This attack is used to obtain passwords. Most password systems do not store plaintext passwords or encrypted passwords. They avoid encrypted passwords because a compromised key leads to the compromise of all passwords in the data store. Lost keys mean that all passwords are invalidated.

Countermeasures to prevent dictionary attacks include:

i. *Delayed response*: Given a login-name/password pair the server provides a slightly delayed yes/no answer. This should avert an attacker from scrutiny adequately many passwords in a realistic instance [13].
ii. *Account locking*: Accounts are locked after few unsuccessful login attempts. Like the previous measure, this appraise is premeditated to prevent attackers from scrutiny sufficiently many passwords in a realistic instant [13].
iii. *Use strong passwords* or pass ware that are complex, are not regular words, and contain a mixture of upper case, lower case, numeric, and special characters.
iv. Store non-reversible password hashes in the user store.

d. Cookie Replay Attacks

In this type of attack, the attacker captures the user's authentication cookie using monitoring software and replays it to the relevance to gain access under a counterfeit identity.

Countermeasures to prevent cookie replay include:

a) Use an encrypted communication channel provided by SSL whenever an authenticated cookie is transmitted.
b) Use a cookie timeout to a value that forces authentication after a relatively short time interval. Although this doesn't prevent replay attacks, it reduces the time interval in which the attacker can replay a request without being forced to re-authenticate because the session has timed out.

e. Credential Theft

If your appliance implements its own storage for user's account names and passwords and compare its security to the credential stores provided by the platform. Browser history and cache memory also store each and every user's login information for the future purpose. If the mortal is accessed by someone other than the user who logged on, and the same page is hit, the saved login information will be available.

Countermeasures to help prevent credential theft include:

a) Use and enforce strong passwords.
b) Store password verifiers in the form of one way hashes with added salt.
c) Enforce account lockout for end-user accounts after a set number of retry attempts.
d) To counter the possibility of the browser cache allowing login access, create functionality that either allows the user to choose to not save credentials, or force this functionality as a default policy.

C. Authorization

Based on user identity and task association, authorization to a particular resource or service is either allowed or denied. Top threats that exploit authorization vulnerabilities include [8]:

a. Elevation of Privilege

When you design an authorization model, you must consider the threat of an attacker trying to elevate privileges to a powerful account such as a member of the local administrators group or the local system account. By doing this, the attacker is able to take complete control over the application and local machine.

i. The main ***countermeasure*** that you can use to prevent increase of dispensation is to use least fortunate process, tune, and user accounts.

b. Disclosure of Confidential Data

The disclosure of confidential data can occur if insightful data can be viewed by unauthorized users or client. Confidential data includes application specific data such as credit card numbers, employee details, and financial records and so on. To prevent the disclosure of confidential data you should secure it in persistent stores such as databases and configuration files, and during transit over the network. Only authenticated and authorized users should be able to access the data that is authorized to them. But to Access the system level configuration data should be restricted to administrators.

a) *Countermeasures* to prevent disclosure of confidential data include:

i. Perform role checks before allowing access to the operations that could potentially reveal sensitive data.
ii. Use strong ACLs to secure Windows resources.
iii. Use standard encryption to store sensitive data in configuration files and databases.

c. Data Tampering

Data tampering refers to the unauthorized modification of data.

Countermeasures to prevent data tampering include:

i. Use strong access controls to protect data in persistent stores to ensure that only authorized users can access and modify the data.
ii. Use role-based security to differentiate between users who can view data and users who can modify data.

d. Luring Attacks

A luring attack occurs when an entity with few privileges is able to have an entity with more privileges perform an action on its behalf.

To counter the threat, you must restrict access to trusted code with the appropriate authorization. Using .NET Framework code access security helps in this respect by authorizing calling code whenever a secure resource is accessed or a privileged operation is performed.

D. Network Threats and Countermeasures

The primary components that make up your network infrastructure are routers, firewalls, and switches. They act as the gatekeepers guarding your servers and applications from attacks and intrusions. An attacker may exploit poorly configured network devices. Vulnerabilities include weak default installation settings, wide open access controls, and devices lacking the latest security patches. Top network level threats include:

a. Information Gathering

Information gathering can reveal detailed information about network topology, system configuration, and network devices. An attacker uses this information to mount pointed attacks at the discovered vulnerabilities.

a) Vulnerabilities –

Common vulnerabilities that make your network susceptible to an attack include:

i. The inherently insecure nature of the TCP/IP protocol suite
ii. Configuration information provided by banners
iii. Exposed services that should be blocked

b) Attacks –

Common information-gathering attacks include:

i. Using **Tracert** to detect network topology.
ii. Using **Telnet** to open ports for banner grabbing.
iii. Using port scans to detect open ports.
iv. Using broadcast requests to itemize hosts on a subnet.

c) Countermeasures to prevent information gathering include:

i. Configure routers to restrict their responses to foot printing requests.
ii. Configure operating systems that host network software (for example, software firewalls) to prevent foot printing by disabling unused protocols and unnecessary ports.
iii. Use generic service banners that do not give away configuration information such as software versions or names.
iv. Use firewalls to mask services that should not be publicly exposed.

b. Sniffing

Sniffing or *eavesdropping* is the act of monitoring traffic on the network for data such as plaintext passwords or configuration information. With a simple packet sniffer, an attacker can easily read all plaintext traffic. Also, attackers can crack packets encrypted by lightweight hashing algorithms and can decipher the payload that you considered to be safe. Packet sniffing is a passive attack where packets are read from a network stream. A packet sniffer is a piece of software or hardware that monitors all the traffic across the network. Traffic monitoring tools are tcpdump/windump, Ethereal, and many more [15].

a) Vulnerabilities –

Common vulnerabilities that make your network susceptible to data sniffing include:

i. Weak physical security
ii. Lack of encryption when sending sensitive data
iii. Services that communicate in plain text or weak encryption or hashing
iv. Failure to encrypted data [14].

b) Attacks –

The attacker places packet sniffing tools on the network to capture all traffic.

c) Countermeasures to help prevent sniffing include:

i. Use strong physical security and proper segmenting of the network. This is the first step in preventing traffic from being collected locally.
ii. Encrypt communication fully, including authentication credentials. This prevents sniffed packets from being usable to an attacker. SSL and IPSec (Internet Protocol Security) are examples of encryption solutions.
iii. Strong physical security that prevents rogue devices from being placed on the network.
iv. Encrypted credentials and application traffic over the network.
v. Some techniques can be used to determine whether the NIC (Network Interface Card) on the suspect machine is running in promiscuous mode or not [15].

c. Spoofing

Spoofing, also called *identity obfuscation*. Spoofing is a means to hide one's true identity on the network. To create a spoofed identity, an attacker uses a fake source address that does not represent the actual address of the packet. Spoofing may be used to hide the original source of an attack or to work around network access control lists (ACLs) that are in place to limit host access based on source address rules.

a) Vulnerabilities –

Common vulnerabilities that make your network susceptible to spoofing include:

i. The inherently insecure nature of the TCP/IP protocol suite
ii. Lack of ingress and egress filtering. Ingress filtering is the filtering of any IP packets with untrusted source addresses before they have a chance to enter and affect your system or network. Egress filtering is the process of filtering outbound traffic from your network.

b) Attacks –

An attacker can use several tools to modify outgoing packets so that they appear to originate from an alternate network or host.

2.4.3.3 Countermeasures to prevent spoofing include:

i. Filter incoming packets that appear to come from an internal IP address at your perimeter.
ii. Filter outgoing packets that appear to originate from an invalid local IP address.
iii. You can use ingress and egress filtering on perimeter routers.

d. Session Hijacking

Also know as man in the middle attack. The attacker host deceives a server or client host so that it appears to be the desired destination.

a) Vulnerabilities –

Common vulnerabilities that make your network susceptible to session hijacking include***:***

i. Weak physical security
ii. The inherent insecurity of the TCP/IP protocol suite
iii. Unencrypted communication

b) Attacks –

An attacker can use several tools to combine spoofing, routing changes, and packet manipulation**.**

c) Countermeasures to help prevent session hijacking include:

i. Use encrypted session negotiation.
ii. Use encrypted communication channels.
iii. Stay informed of platform patches to fix TCP/IP vulnerabilities, such as predictable packet sequences.
iv. Session encryption
v. Stateful inspection at the firewall

e. Denial of Service

Denial of service denies legitimate users access to a server or services. The SYN flood attack is a common example of a network level denial of service attack. It is easy to launch and difficult to track. The aim of the attacker is to send more and more requests to a server that it can handle. Network-layer denial of service attacks usually try to deny service by flooding the network with traffic, which consumes the available bandwidth and resources.

a) Vulnerabilities-

That increase the opportunities for denial of service include:

i. The inherent insecurity of the TCP/IP protocol suite
ii. Weak router and switch configuration
iii. Unencrypted communication
iv. Service software bugs

b) Attacks –

Common denial of service attacks include:

i. Brute force packet floods, such as cascading broadcast attacks
ii. SYN flood attacks
iii. Service exploits, such as buffer overflows

c) Countermeasures –

to prevent denial of service include:

i. Apply the latest service packs.
ii. Harden the TCP/IP stack by applying the appropriate registry settings to increase the size of the TCP connection queue, decrease the connection establishment period, and employ dynamic backlog mechanisms to ensure that the connection queue is never exhausted.
iii. Use a network Intrusion Detection System (IDS) because these can automatically detect and respond to SYN attacks.
iv. Filtering broadcast requests
v. Filtering Internet Control Message Protocol (ICMP) requests
vi. Patching and updating of service software

III. THREATS MODELING PRINCIPLES

Threat modeling at the design phase is really the only way to bake security into the *SDLC*- Michael Howard, Microsoft [17].

Threat modeling should not be a onetime only process. It should be an iterative process that starts during the early phases of the design of your application and

continues throughout the application life cycle. There are two reasons for this. First, it is impossible to identify all of the possible threats in a single pass. Second, because applications are rarely static and need to be enhanced and adapted to suit changing business requirements, the threat modeling process should be repeated as your application evolves [16]. Threat modeling is also used to refer, variously, to analysis of software, organizational networks or systems, or, as in [18] even industrial sectors.

A. The Process

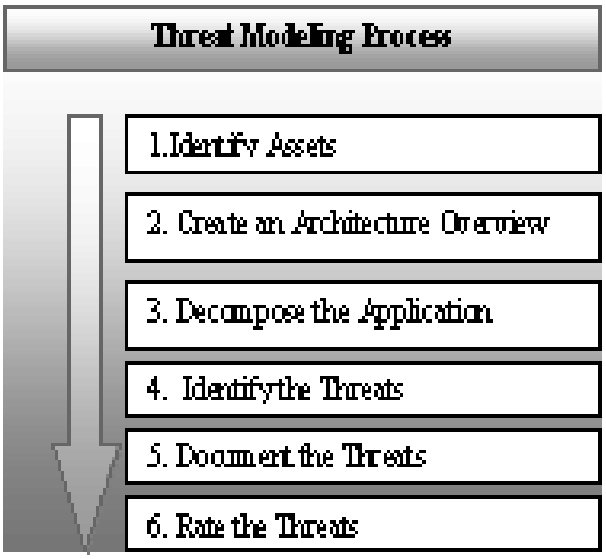

Figure .1 shows the threat modeling process that you can perform using a six-stage process [17].

A. An Overview of the Threat Modeling Process [19][20]

a) Identify assets –

Identify the valuable assets that your systems must defend.

b) Create an architecture overview –

Use simple diagrams and tables to document the architecture of your application, including subsystems, trust boundaries, and data flow.

c) Decompose the application –

Decompose the architecture of your application, including the underlying network and host infrastructure design, to create a security profile for the application. The aim of the security profile is to uncover vulnerabilities in the design, implementation, or deployment configuration of your application.

d) Identify the threats –

Keeping the goals of an attacker in mind, and with knowledge of the architecture and potential vulnerabilities of your application, identify the threats that could affect the application.

e) Document the threats –

Document each threat using a common threat template that defines a core set of attributes to capture for each threat.

f) Rate the threats –

In an organization where threat and vulnerability management is governed by unyielding risk an administration principle, the following formula is typically used to assign a risk score to a threat [21]:

Risk = Probability of Occurrence x Business Impact

Firstly Rate the threats to priority wise and address the most significant threats. These threats in attendance the biggest risk. The rating process weights the probability of the threat against damage that could result should an attack occur. It might turn out that certain threats do not necessitate any action when you compare the risk posed by the threat with the momentous mitigation costs [22].

IV. SECURITY SOLUTION TO DEFEND THREATS

A. Security through Cryptography –

Cryptography is the process to convert or to encrypt the information into the unreadable format. Even if the intruder accesses the data, it should not be able to understand the content of it. Cryptography can be symmetric (which uses same key to encrypt and decrypt the data) and asymmetric (which uses one key to encrypt and other to decrypt the data). This security preserves the integrity and confidentiality of data. Techniques like MD5 (Message Digest 5), Digital Signature, SHA (Secure Hash Algorithm), MAC (Message Authentication Codes) are used to preserve the security principles.

B. Security through TTP (Trusted Third Party)-

This service comes in picture when the security to the message in communication channel is provided by the some third party which can be trusted. A common example can be Public Key infrastructure (PKI) [22].in which a trusted third party like Certifying Authority (CA) issues a certificate to the legitimate clients for authenticating them. This preserves authentication security principle.

C. Security Through IDS (Intrusion Detection Systems)-

Intrusion Detection System [23][24] in communication network monitors the node for malicious behavior. Anomaly based IDS is used in such process where any incongruity in the network confirms an attack. Profiles are maintained in the database of IDS which is the normal behavior of a node. These profiles are made under training period. Such profiles can either be static or dynamic in nature. IDS can be designed inside the node or can even work as TTP.

D. Ddos Attack Prevention

Countermeasures for defending against DoS are defined by the class *'DoSDefence'* that consists of several subclasses. Actually, there is no fundamental defense against DDoS and DRDoS attacks. The TCP SYN attacks can be mitigated (the class *'TCPSYNDefence'*) by increasing the size of the SYN ACK queue, decreasing the time-out waiting for the three-way handshake, and installing software patches. Other defenses are not presented here due to limited space [25][26].

a) Monitoring configuration and attack status using the CLI [27]

b) Debugging attack status and suspect packets using the CLI [27]

c) Mirroring suspect traffic to a mirror port for detailed analysis [27]

d) Logging attacks to the Logging Facility [27]

e) SNMP traps [28]

E. Security Through Other Methods-

Several models and algorithms have been proposed which assures in detecting and preventing the malicious behavior of nodes. Such methods constituent the concept of above security solutions like cryptography, certification system, intrusion detection system etc.

a) Data-level Authorization: -

Web Services that are exposed through the corporate firewall are liable to be attacked if they are not properly secured with authentication and access control measures. Administrators need to create and enforce policies that guard against unauthorized use of Web Services by allowing only trusted authorized consumers permission to read, write or alter data according to predefined privileges [28].

b) Data Validation: -

Data being carried via SOAP messages can be headed to multiple applications. Some may not even perform the most basic error checking. However, even when proper data validation is performed within the application-layer, the large volume of SOAP message flows still needs to be consistently "sanity checked" for malicious or inappropriate content. This, very rapidly, becomes a burden to the back-office system [28].

V. REFERENCES: -

[1]. http://en.wikipedia.org/wiki/Security

[2]. U.S. Government Printing Office Office of Information Dissemination Program Development Service ,"Authentication".

[3]. http://en.wikipedia.org/wiki/STRIDE_(security)

[4]. http://www.gpoaccess.gov/authentication/authenticationwhitepaperfinal.pdf,October 13, 2005.

[5]. Marwan Abi-Antoun, Je_rey M. Barnes," STRIDE-based security model in Acme", http://reports-archive.adm.cs.cmu.edu/anon/isr2010/CMU-ISR-10-106.pdf, January 2010,CMU-ISR-10-106

[6]. Tom Olzak,"A Practical Approach to Threat Modeling," http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A_Practical_Approach_to_Threat_Modeling.pdf" , March 2006

[7]. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan," Building Secure Web Services", http://msdn.microsoft.com/en-us/library/ff648643.aspx, Published: June 2003

[8]. "Security of Rich Internet Applications", http://www.magicsur.com.uy/userfiles/file/Documentos/Security-of-RIA-10-08.pdf,

[9]. October 2008

[10] Michael Howard, James A. Whittaker,"Demystifying the Threat-Modeling Process", http://see.xidian.edu.cn/hujianwei/papers/022-Demystifying%20the%20threat%20modeling%20process.pdf, Published by the IEEE Computer Society, 1540-7993/05/$20.00 © 2005 IEEE

[11]http://www.24x7code.com/main/authentication.aspx

[12] ICS-CERT - Control Systems Analysis Report,"http://www.us-cert.gov/control_systems/pdf/ICS-CERT%20CSAR-SSH%20SCANNING.pdf", April 24, 2010, pp.1-4

[13]http://en.wikipedia.org/wiki/Application_security

[14] Benny Pinkas, Tomas Sander,"Securing Passwords Against Dictionary Attacks", http://www.pinkas.net/PAPERS/pwdweb.pdf, pp.1-2

[15] OWASP," The Open Web Application Security Project", https://www.owasp.org/index.php/Sniffing_application_traffic_attack

[16] Ahmed Obied," An Analysis of Network Attacks and their Countermeasures", http://ahmed.obied.net/research/papers/bsc_thesis.pdf,April 15, 2005

[17] Adam Shostack, Experiences Threat Modeling at Microsoft, http://www.homeport.org/~adam/modsec08/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf

[18] Nathan Sportsman, Founder and Chief Executive Officer," http://www.praetorian.com/presentations/Praetorian_Threat_Modeling_Presentation.pdf", Entire contents © 2011 Praetorian

[19] Craig Rubens, Cleantech Terror Alert,"Hacking the Grid",Earth2Tech, http://earth2tech.com/2008/06/26/cleantech-terror-alert-hacking-the-grid/, June 26, 2008,

[20] Ben hickman, VP Engineering,"Application Security and Threat Modeling", http://cpd.ogi.edu/seminars04/hickmanthreatmodeling.pdf

[21] Adam Shostack," Reinvigorate your Threat Modeling Process", http://msdn.microsoft.com/en-us/magazine/cc700352.aspx, 2008 july

[22] Tom Olzak ,"A Practical Approach to Threat Modeling", http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A_Practical_Approach_to_Threat_Modeling.pdf" , March 2006

[23] Lioyd,S. and Adams,C."Understanding PKI: Concepts, Standards and Deployment Considerations" ISBN 0-672-32391-5

[24] "Intrusion Detection System" http://www.intrusiondetection-system-group.co.uk/, Link visited on December 2010

[25] Sahu, S and Shandilya, S K - A Comprehensive Survey On Intrusion Detection In Manet, International Journal of Information Technology and Knowledge Management, Volume 2, No. 2, pp. 305-310July-December 2010

[26] Artem Vorobiev, Jun Han, and Nargiza Bekmamedova ,"An Ontology Framework for Managing Security Attacks and Defences in Component Based Software Systems", 19th

Australian Conference on Software Engineering, 1530-0803/08 $25.00 © 2008 IEEE DOI 10.1109/ASWEC.2008.25, March 23,2010

[27] maureen, November 21st, 2009, http://www.netfitz.com/ddos-attack-prevention-the-best-medicine/

[28] http://www.alliedtelesis.com/media/fount/software_reference/291/at8600/dos.pdf

[29] www.forumsystems.com, Anatomy o f a Web Services Attack," A Guide to Threats and Preventative Countermeasures".