An Approach to Counteracting the Common Cyber-attacks According to the Metric-Based Model

Mohammad Sirwan Geramiparvar, Nasser Modiri

MS Candidate in Computer Engineering (Software), Faculty of Electrical, Computer and IT Engineering, Islamic Azad University of Zanjan

Sirwan_gp@yahoo.com

Ph. D, Assistant Professor in Faculty of Electrical, Computer and IT Engineering, Islamic Azad University of Zanjan

Abstract

Nowadays the words such as cyber and electronic attacks are heard of more than any other ones. They refer to the attacks which can ruin the structure and foundation of a company, organization, or even a ministry and cause irreparable damages. Analyzing these intrusions, we ascertain that every cyber-attack benefits from four general weaknesses: innate and structural weakness, configurations, design and implementation, and human-made errors. Concentrating on the relevant weaknesses and overcoming them, we can achieve a secure and stable system. Studying and analyzing common attacks such as XSS and SQL Injection and using metric-based model, this research, therefore, aims to focus on the weaknesses and to present approaches to overcoming the weaknesses and counteracting these attacks.

Keywords: Metric-Based Model, Cyber-attacks, General Weaknesses, Countermeasures, XSS, SQL Injection

I. Introduction

Nowadays intruders and electronic thieves (scammers) use various methods to have unauthorized access to personal, confidential and important information or bank accounts. In fact, the wars of hostile states have turned into electronic and cyber wars. More of these attacks occur every day. According to the statistics provided by security companies, over 140 million new and zero-day malwares entered the cyber world in 2014[1]. Also, the success of these malwares and attacks indicate the inefficiency of conventional methods for detecting and counteracting these attacks and discovery of new methods for abuse by intruders. Therefore, it is more essential now than before to generate new methods in order to detect and counteract cyber and malware attacks. According to the statistics provided by reliable companies, two common attacks which occur more than other ones are XSS and its types and SQL Injection (Figure 1).

[1] http://www.av-test.org/en/statistics/malware/?=

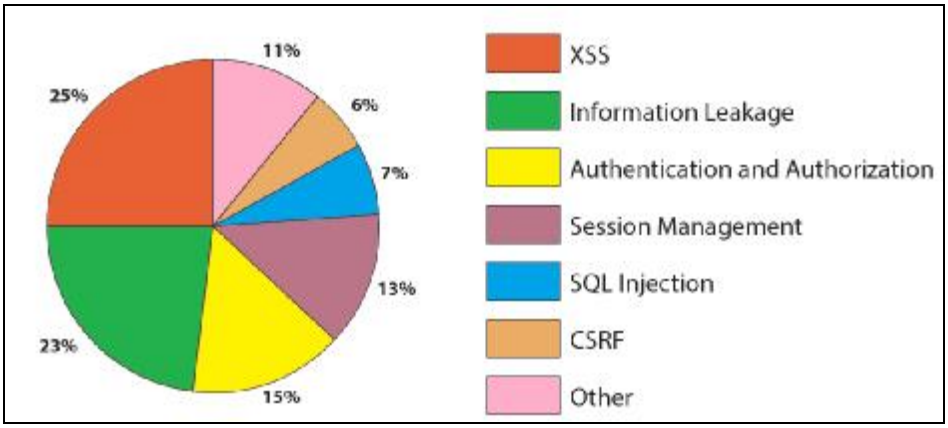

Figure. 1. Statistics Pertaining to the Majority of Security Breaches on Websites in 2013[2]

In such attacks, the intruder or malware can take advantage of all 4 types of weaknesses (structural, configuration, design, human-made) and completes the attack stages so that the objectives can be achieved.

In the investigation of identifying and classifying computer malwares, all the academic studies and researches fall into three categories of signature-based methods, behavior-based and heuristic ones. Each method has been separately simulated in a virtual environment such as Virtual Machine. They are analyzed by monitoring the behavior and performance with a dynamic method or according to the physical characteristics and system behaviors (modification, deletion, and so on) using static analysis. However, there are 2 malware sources and 3 well-known models in the process investigation of attacks and malwares. Malware sources are the ontological model of Swimmer (Swimmer, 2008) and MAEC model which belongs to MITRE ™ Corporation. Regarding the process models, the first model, which was presented in 1998, belongs to Howard and Longstaff (Howard et al., 1998). Using a specific flow graph in a simple and fluent language, it investigates an attack in a process way. They divided the attack process in 7 phases. The intruders (hackers, spies, and so on) are placed in the first phase, while the tools such as instructions, information exchange and so on which are used by them are put in the second phase. In the third phase, there are threats which exist, such as poor design, implementation, or structuring. The actions which are taken for the attacks are placed in the fourth phase. The instances are sniffing, reading, copying, and so on. The fifth phase is regarding the attack objectives such as data, bank accounts, and so on. The next phase contains the unsupervised results such as denial of service, resource theft, and so forth. Finally, the seventh phase includes attack objectives such as destruction, political purposes, financial reasons, and so on. Although the above-mentioned model is very simple and rudimentary, it has inspired many subsequent models which used it. The next model which is studied was introduced by Gadelrab and affiliates (Gadelrab et al., 2007). It has 5 dimensions. These 5 dimensions of malware process are as follows: intrusion source (remote or internal), acquiring or increasing the privilege (root, system, user, ordinary), vulnerability (configuration, implementation), carrier (network traffic or a normal action), and objective (operating system, memory, and so on). They tested and implemented their model in 8 states in the laboratory. The results of their evaluation obtained by IDSs indicated the efficiency or success of the model; however, this model was then improved by Saber and co-workers (Saber et al., 2010), and a better model was presented. The second model was presented by Gadelrab and fellows (Gadelrab et al., 2008). Testing 39 samples of

[2] http://www.cenzic.com/ downloads/Cenzic_Vulnerability_Report_2014.pdf

malwares analyzed on CME list (Common Malware Enumeration), they evaluated the executive patterns for analyzed attacks. They realized that 8 steps could be considered for the attack. The attack steps are as follows: Reconnaissance (R), Victim Browsing (VB), Execute Program (EP), Gain Access (GA), Implant Malicious Code (IMC), Compromise Data Integrity (CDI), Denial of Service (DOS), and Hide Trace (HT). According to these 8 steps, they analyzed some well-known malwares such as Code Red-I, Code Red-II, Sasser, Trinoo, and so on. It is worth mentioning that this model only deals with the attack process. It does not implement the details and attack implementation commands such as buffer rewriting or code sequence. For instance, the sequence of phases in the attack by Cod Red-I are as follows: GA, IMC, EP, EP, VB, CDI, EP, VB, DOS. Using the second model presented by Gadelrab and studying a large number of CWEs (Common Weakness Enumeration) and CAPECs (Common Attack Pattern Enumeration & Characterization), Geramiparvar and Modiri introduced 9 metrics as measures for identification and classification of computer malwares and cyberattacks (Geramiparvar et al., 2014). Also, they prioritized the metrics. Using fuzzy AHP (Analytical Hierarchy Process) and according to the impact of each metric in establishing the attack step or their effectiveness in 8 attack phases, they calculated the weights and importance pertaining to each metric and introduced their metric-based model.

This paper consists of 4 sections. In the second section, the attacks are introduced and explained according to the metric-based model, and then they are analyzed. Identification and prevention measures are introduced and the impacts of metrics on attack stages are investigated and then the approach is proposed. In the last part, the discussion and conclusion are presented. Finally, some suggestions are made for future works and relevant researches in the eighth part.

II. Explanation and analysis of XSS and SQL Injection Attacks

Now, to learn more about common attacks, we introduce XSS and SQL injection and analyze them in details.

A. SQL Injection Attack and Phase Analysis

Regarding SQL Injection attack and the quality of its phases, it can be stated that the malware or intruder requires an initial reconnaissance first in order to access database tables, databases, table names and its fields. This reconnaissance can be done by sending a misleading query, mismanaging the errors and so on. After that, it starts browsing the database and acquiring the username and password in order to enter the database. After acquiring the username, password, and the necessary privileges, it starts injecting codes and sending search queries in order to acquire information. Finally, it can modify or delete data after acquiring the necessary information and enough authorizations (upgrading the privileges). According to the second model by Gadelrab, attack phases, therefore, can be summarized as R→VB→GA→IMC→CDI...

B. XSS Attack, Types and Analysis

XSS attacks are a form of injection attacks in which malicious scripts are injected into trusted or harmless websites. The weaknesses which result in the success of these attacks are very extensive. These attacks occur in web-based applications in which user login is used and the output is generated without authentication and encryption. The intruder uses XSS in order to send malicious scripts to an unsuspicious user. The browser of the end user has no knowledge to recognize that the script should not be trusted, so it executes the script. Because it thinks that the

script comes from a reliable source. The malicious script can access to all cookies, session tokens, or every other piece of important information which is kept by the browser on the website. These scripts can even rewrite the contents of HTML pages. XSS attacks occur at two times:

•When data enters a web-based application through an unreliable source, the highest frequency pertains to the web request.

•When data included in a dynamic content which is sent to a web user are not verified to find the malicious content.

XSS attacks fall into two categories named reflected and stored; however, there is also a less-known third category which is named DOM-based attacks. Assume that a user logs on to his bank account. Now an intruder can send him an email which contains intriguing information and deceive him into clicking on a link in the email (GA). The link contains a script or refers to a malicious script (IMC). Through an iFrame, it probably deceives a real user into performing a subversive action on a registration or approval form (such as funds transfer from the victim account). Then the intruder can paste the built-in scripts and executes desirable operations (EP) with user's authorization. The reconnaissance phase may sometimes occur before these actions: R→GA→IMC→EP...

C. Introduction of Identifying and Preventing Metrics

As explained in part I, Geramiparvar and Modiri (Geramiparvar et al., 2014) proposed 9 metrics as measures to identify and classify the malwares and cyber-attacks. Each metric was introduced in response to many weaknesses of which malwares and attacks take advantage so that they can perform the next phases of the intrusion or attack. These metrics were extracted after studying a large number of CWEs. They are explained in Table I:

TABLE I
THE PROPOSED METRICS FOR IDENTIFICATION AND CLASSIFICATION OF MALWARES

Metric	Description
Input validation	How do we know that the input data is valid and reliable? Indeed, it includes how your service filters, scrubs, and rejects inputs and outputs before additional processing. For example restricting inputs from specific entry points or by special formats and encoding outputs in exit points. Can we rely on data obtained from databases or file shares?
Authentication	What is your identity? Authentication is the process in which an entity proves the identity of another entity, typically through credentials, such as a username and password.
Authorization	What can you do? Authorization is the means by which your service providers access controls for resources and operations
Installation and configuration management	Who will run your application? Which one of databases are you connected to? How to make these preferences secure? Configuration Management refers to how your service handles database connection, administration, and other configuration settings.
Sensitive Data	How applications protect sensitive data? What applications

	are dealing with is called sensitive data which must be preserved in the memory, network, or database.
Session Management	How applications manage and protect client sessions? A session consists of a series of interactions between a client and your service
Cryptography	How do you protect credential data (privacy)? How do you prevent data and libraries from being distorted (integrity)? How to provide initial values to generate random variables that must be encrypted strongly? In fact cryptography deals with maintaining integrity and confidentiality.
Exception Management	What do you do if an application or calling for a stored procedure fails? How many sensitive system level details would be disclosed? Does it return error information directly? Exception management is about error management and error return as well as controlling exceptions occurred applications are running.
Auditing and Logging	It describes what action was done by anyone at any given time. Auditing and logging refer to how security-related events are recorded, monitored, and audited.

To understand these metrics better and ascertain which weakness the malwares (or the intruders) take advantage of so that they continue their attacks, a list of vulnerabilities and threats caused by metrics is indicated in Table II. The list also helps understand how failure to comply with these metrics properly helps complete the phases of attacks.

TABLE II
VULNERABILITIES RESULTING FROM FAILURE TO COMPLY WITH THE METRICS PROPERLY

Metrics	Vulnerabilities
Input Validation	• Using non-validated input in the HTML output • Using non-validated input used to generate SQL Queries, relying on client side validation • Using input file names, URLs, user names for security decisions. • Using application-only filters for malicious inputs. • Looking for known bad patterns of input. • Trusting data read from database, files shares and other network resources. • Failing to validate input from sources including cookies, Query string parameters, HTTP headers, database, and other network resources.
Authentication	• Using weak passwords. • Storing clear text credentials in the configuration files. • Passing clear text credentials over the network. • Permitting prolonged session lifetime. • Mixing authentication with personalization (such as name and date of birth, etc.).
Authorization	• Relying on a single gatekeeper.

	• Failure to lock down system resources against application identities. • Failure to limit database access to specific stored procedures. • Using inadequate separation of privileges. Permitting over privileged accounts
Configuration and Installation Management	• Using insecure custom administration interfaces. • Failing to secure configuration files on the server. • Storing sensitive information in the clear text. • Having too many administrators. • Use over privileged process accounts and service accounts. • Installing from unknown/untrusted sources.
Sensitive data	• Storing secrets when you do not need to. • Storing secrets in code. • Storing secrets in clear text in files, registry, or configuration. • passing sensitive data in clear text over the networks.
Session Management	• Passing session IDs over unencrypted channels. • Permitting prolonged session lifetime. • Having insecure session state stores. • Placing session identifiers in query strings.
Cryptography	• Using custom cryptography. • Using the wrong algorithm or a short key. • Managing or storing keys insecurely. • Overusing a key over the long term. • Distributing keys insecurely.
Exceptions Management	• Failure to use structured exception Handling. • Revealing too much information to the client.
Auditing and Logging	• Failing to audit failed logons. • Failing to secure log files. • Failure to audit across application tiers.

Likewise, in order to identify the best metrics and ascertain which consequences we face if each metric is not complied with properly, a list of attacks and threats is indicated in Table III. Some of these attacks may also be caused by ignoring two or more metrics, so the metric which can be introduced as the main reason for the attack is mentioned here.

TABLE III
THE LIST OF ATTACKS DUE TO IGNORING THE METRICS

Metric	Attacks and Threats
Data validation	• Buffer Overflows • XSS or Cross-site scripting • SQL Injection • canonicalization attacks • The query strings • The form fields • Cookie manipulation

	• The HTTP headers
Authentication	• Network eavesdropping • Attacks using client awkwardness (universal search) • Dictionary attack • Cookie replay attack • Credential theft
Authorization	• Elevation of privilege • disclosure of confidential data • luring attacks
Configuration and Installation Management	• Unauthorized access to custom administration interfaces • Unauthorized access to configuration store • Retrieving Clear text configuration codes • Lack of single, specific auditing and accounting
Sensitive data	• access to sensitive data on storage media • access to sensitive data in memory (including Process dumps) • Eavesdropping Network • Information disclosure
Session Management	• Session hijacking • Session replay • Man-in-the-middle attacks
Cryptography	• Theft of decryption keys • Password cracking
Managing exceptions	• Disclosure of sensitive data or applications' details • Denial of Service (DoS)
Auditing and Logging	• Repudiation • An attacker abusing a program with leaving no trace • Attacker hides his/her trace

D. The Impact of Metrics on the Phases of Attacks and Recommendations

After explaining XSS and SQL Injection attacks in the previous part and introducing the metrics, each metric and its probable role in attack phases is investigated in Table IV:

TABLE IV
THE LIST OF METRICS' ROLES IN INVESTIGATED ATTACKS

Metric	SQL Injection	XSS
Data validation	Using non-validated inputs (IMC)	1. Using non-validated inputs in the HTML outputs (GA, IMC) 2. Trusting read data from database

		resources ... (IMC) 3. Failing to validate input from all sources, including cookies... (IMC)
Authentication	Permitting over privileged accounts (GA)	1. Passing clear text credential over the network (R) 2. Permitting prolonged session lifetime (R)
Authorization	1. Failing to limit database access (CDI) 2. Failing to lock down system resources (CDI)	Failing to limit database access to specified stored procedures (GA, EP)
Configuration and Installation Management	1. Storing configuration data in clear text (GA, CDI) 2. Failing to secure configuration files on the server (VB)	Using over privileged process accounts and service accounts (EP)
Sensitive data	-	-
Session Management	Permitting prolonged session lifetime (R)	Placing session identifiers in query strings (R, GA)
Cryptography	Not encrypting messages (GA)	Not encrypting message (GA)
Exception Management	Revealing too much information to the client (R)	-
Auditing and Logging	Failing to audit failed logons (GA)	-

III. A Solution Based on Metrics

As a proposed method and a solution to prevent common cyber-attacks and given the fact that all the weaknesses resulting from the metrics cause security holes so that attack phases are completed this way, a list of actions are mentioned as countermeasures in Table V. Complying with these points, a secure and stable system can be generated and other similar intrusions can be prevented.

TABLE V
COUNTERMEASURES AGAINST CYBER-ATTACKS AND MALWARES

Metric	Countermeasure
Data validation	• Do not trust input, consider centralized input validation. • Do not rely on client-side validation. • Be careful with canonicalization issues. • Constrain, reject, and sanitize input. • Validate for type, length, format, and

	range.
Authentication	• Partition site by anonymous, identified, and authenticated area. • Use strong passwords. • Support password expiration periods and account disablement. • Do not store credentials (use one-way hashes with salt). • Encrypt communication channels to protect authentication tokens. • Pass Forms authentication cookies only over HTTPS connections.
Authorization	• Use least privileged accounts. • Consider authorization granularity. • Enforce separation of privileges. • Restrict user access to system-level resources.
Configuration and Installation Management	• Use least privileged process and service accounts. • Do not store credentials in plaintext. • Use strong authentication and authorization on administration interfaces. • Do not use the LSA. • Secure the communication channel for remote administration. • Avoid storing sensitive data in the Web space.
Sensitive data	• Avoid storing secrets. • Encrypt sensitive data over the wire. • Secure the communication channel. • Provide strong access controls on sensitive data stores. • Do not store sensitive data in persistent cookies. • Do not pass sensitive data using the HTTP-GET protocol.
Session Management	• Limit the session lifetime. • Secure the channel. • Encrypt the contents of authentication cookies. • Protect session state from unauthorized access.
Cryptography	• Do not develop your own. • Use tried and tested platform features.

	• Keep unencrypted data close to the algorithm. • Use the right algorithm and key size. • Avoid key management (use DPAPI). • Cycle your keys periodically. • Store keys in a restricted location.
Managing exceptions	• Use structured exception handling. • Do not reveal sensitive application implementation details. • Do not log private data such as passwords. • Consider a centralized exception management framework.
Auditing and Logging	• Identify malicious behavior. • Know what good traffic looks like. • Audit and log activity through all of the application tiers. • Secure access to log files. • Back up and regularly analyze log files.

IV. Conclusions

In addition to attack phases and their sequential order, the metric-based model informs us of which weaknesses the malwares and cyber-attacks take advantage of and with which metric they take attack steps. Also, the above-mentioned model provides us with a better answer in order to find the similar categories and malwares. Using the results of this model, we can realize that some weaknesses are innate, some are structural, and some others pertain to the system. However, human errors and weaknesses should not be ignored. According to the metric-based model, some of the notable characteristics of malwares are as follows:

•This model can be applied as a language (syntax) in order to express the behavioral characteristics and signature of malwares. In fact, it can act as a new language to declare malware patterns and to explain their performances.

•According to the latest standards and proposed methods, the description of malwares (based on MITRE standard) benefits from dictionaries and rich databases such as CWE, CVE, and CAPECs.

•It can be used in software security systems such as antiviruses, firewalls, Pentest productions, and hardware productions such as UTMs and also in IDSs and IPSs.

•Covering the weaknesses and taking countermeasures, the proposed approach based on this model deals with the details and weaknesses in order to prevent the security holes.

•Given the metrics such as configuration management, installation, access control, surveillance, and so on, it is capable of implementation and compatible with implementation controls of Information Security Management Systems (ISMS).

A. Future Works

With the increasing complexity and combination of extensive attacks, the conventional and common methods are not accountable nowadays, and it appears essential to propose new models

and methods with more capabilities. Other problems such as topless and bottomless malwares are unsolved nowadays because their beginning and ending are not specified. An intruder may launch an attack or malware from a country or point and stop it due to a bug or to avoid attracting attention, and then another intruder continues the attack from another point, or the attack phase and stage may not be specified. All of these cases can be the subject of future researches or works which can be invested on. The human factor is also discussed as a challenge and GAP in the security problems and ISMS. It is worth defining the appropriate models in this regard.

References

i. John D. Howard & Thomas A. Longstaff, (1998). A Common Language for Computer Security Incidents, SANDIA Report, SAND 98-8667.

ii. M. Gadelrab, A. Abou El Kalam, Y. Deswarte, (2008). Execution Patterns in Automatic Malware and Human-Centric Attacks. *Seventh IEEE International Symposium on Network Computing and Applications*, Jul. 10-12, Cambridge, Massachusetts, USA

iii. M. Gadelrab, et al. (2007). Defining categories to select representative attack test-cases. *Proceedings of the 2007 ACM workshop on Quality of protection*, Pages 40-42, ACM New York, NY, USA.

iv. M. Saber, T. Bouchentouf, A. Benazzi, M. Azizi, (2010). Amelioration of Attack Classifications for Evaluating and Testing Intrusion Detection System. Journal of Computer Science, Vol (6), Issue. 7, pp. 716-722.

v. M. Sirwan Geramiparvar, N. Modiri, (2014). Presenting a Metric-Based Model for Malware Detection and Classification. *International Journal Of Computer & Information Technology (IJOCIT)*, Vol (2), Issue. 4, pp. 528-539.

vi. Swimmer, M. towards an Ontology of Malware Classes. *[Online]* http://www.scribd.com/doc/24058261/Towards-an-Ontology-of-Malware-Classes