Interested in learning more about cyber security training?

SANS Institute InfoSec Reading Room

The Best Defenses Against Zero-day Exploits for Various-sized Organizations

Every organization is at risk for zero-day exploits regardless of size. These exploits will often circulate for months until the vulnerability is made public, leaving organizations unprotected. This paper will discuss various methods that organizations can use to better detect zero-day exploits. Organization size will be examined to determine whether it plays a part in the detection methods used regarding zero-day exploits. Information technology professionals will be better informed and therefore, better prepared to d...

The Best Defenses Against Zero-day Exploits for Various-sized Organizations

GIAC (GSEC) Gold Certification

Author: David Hammarberg, dhammarberg@macpas.com
Advisor: Dominicus Adriyanto
Accepted: September 21st 2014

Abstract

Every organization is at risk for zero-day exploits regardless of size. These exploits will often circulate for months until the vulnerability is made public, leaving organizations unprotected. This paper will discuss various methods that organizations can use to better detect zero-day exploits. Organization size will be examined to determine whether it plays a part in the detection methods used regarding zero-day exploits. Information technology professionals will be better informed and therefore, better prepared to defend against zero-day exploits by knowing and using this information in their unique environment.

1. Introduction

Zero-day exploits are vulnerabilities that have yet to be publicly disclosed. These exploits are usually the most difficult to defend against because data is generally only available for analysis after the attack has completed its course. These vulnerabilities are highly sought after by cyber criminals, governments, and software vendors who will pay high prices for access to the exploit (Bilge & Dumitras, 2012).Organizations of every size face similar risks from zero-day exploits, but the defenses used against the threat of zero-day exploits are quite different if they defend against them at all.

These zero-day attacks can take the form of polymorphic worms, viruses, Trojans, and other malware. According to Kaur & Singh (2014) the most effective attacks that avoid detection are polymorphic worms which show distinct behaviors. "This includes: complex mutation to evade defenses, multi-vulnerability scanning to identify potential targets, targeted exploitation that launches directed attacks against vulnerable hosts, remote shells that open arbitrary ports on compromised hosts to connect to at a later time, malware drops in which malicious code is downloaded from an external source to continue propagation" (Kaur & Singh, 2014, p. 95).

There were more zero-day vulnerabilities discovered in 2013 than in any previous year according to Symantec's Internet Security Report of 2014. "The 23 zero-day vulnerabilities discovered represent a 61 percent increase over 2012 and are more than the two previous years combined" (Symantec Corporations, 2014, p. 6).

The research community has broadly classified the defense techniques against zero-day exploits as statistical-based, signature-based, behavior-based, and hybrid techniques (Kaur & Singh, 2014). The primary goal of each of these techniques is to identify the exploit in real time or as close to real time as possible and quarantine the specific attack to eliminate or minimize the damage caused by the attack. Another challenge these methods face is making sure the victim's machine threshold for delay for analysis and quarantine is not exceeded. This may cause destabilization of the attacked machine (Yao, Xiang, Qu, Yu & Gao, 2012).

David Hammarberg, dhammarberg@macpas.com

The statistical-based approach to detecting zero-day exploits in real time relies on attack profiles built off of historical data. This approach does not usually adapt well to changes in zero-day exploit data patterns. Any changes in a zero-day exploit's pattern would require a new profile to be learned by the system (Kaur & Singh, 2014).

Polymorphic worm detection is the primary focus of signature-based detection. Signature-based detection is dependent on signatures made from publically known exploits. These signatures will defend against some variations of the original signature or exploit depending on the process used by the attackers to conceal the original known exploit's signature. This detection method is further broken down into content-based, semantic-based, and vulnerability-based detections (Kaur & Singh, 2014).

Behavior-based model defense is based on the analysis of the exploit's interaction with the target. While often based on analysis data captured using high interaction honeypots, normal interactions can be learned, future activity predicted, and exploits classified into behavior groups. Interactions outside the normal behavior groups would be suspicious and quarantined. This method then has the potential to detect and analyze potential zero-day exploits in real time(Alosefer & Rana, 2011.).

The hybrid detection model combines models previously mentioned using a heuristic approach. This method claims to be stronger against polymorphic, metamorphism, and other obfuscations (Ting, Xiaosong & Zhi, 2009). The hybrid method used will depend on what other methods of detection are combined in the environment.

Organization size will be examined to determine whether it plays a part in the detection methods used regarding zero-day exploits. Three size classifications of organizations will be used in this paper. Small organizations have less than100 computers, medium-sized organizations have 100-1,000computers and large/very large organizations have more than 1,000 computers (Briney & Prince, 2002).

David Hammarberg, dhammarberg@macpas.com

2. Analysis of statistical-based, signature-based, behavior-based, and hybrid detection-based techniques

Every organization connected to the internet, independent of the organization's size, has at least one common threat which is a zero-day exploit. Zero-day exploits are vulnerabilities that have not yet been patched by the vendor of the software containing the vulnerability and are being used or could be used for harmful purposes. The goals of these exploits include, but are not limited to, monitoring of the target's operations, theft of secrets, and production disruption. These exploits are often designed or purchased for those specific purposes by various organizations including governments and organized crime. There is currently more demand in the market for zero-day exploits than there is supply which makes the business of selling these exploits lucrative (Bilge & Dumitras, 2012).

In order for the malicious zero-day exploits to remain valuable and useful the exploit needs to remain undetected by an organization's defense in-depth strategies until after the goal of the attacker has been achieved. The longer the exploit goes undetected, the more lucrative the exploit. The average exploit goes undetected for 312 days permitting the harmful purpose of the exploit to affect many organizations (Bilge & Dumitras, 2012). The malicious zero-day exploit could be an exploitation of a vulnerability left unknowingly or knowingly by the vendor of the application or could be an alteration of the application's original source code by an attacker. Both the vendor and the attacker would use code obfuscation to cloak the vulnerability.

"Code obfuscation is the practice of making code unintelligible, or at the very least, hard to understand. The process of code obfuscation is the application of transformations to the code which changes the physical appearance of the code while preserving the black-box specifications of the program" (Balakrishnan & Schulze, 2005, p. 1). The better an attacker's code obfuscation, the better the application with the exploited vulnerability acts like it should and appears like it should to an organization's defense in-depth strategies. Code obfuscation is used by programmers to hide intellectual property and thwart reverse engineering, and is a similar technique used by attackers to hide malicious code without being detected (Balakrishnan & Schulze, 2005). Regardless

of code obfuscation, given enough time and resources any application can be reverse engineered (Collberg, Thomborson & Low, 1997). There is a finite life time of all zero-day exploits. The closer the life span of a zero-day exploit is to zero, the less time it has to cause damage across various organizations. Once a zero-day exploit has been made public and patches are made available to correct the vulnerability, the exploit is considered preventable and technically is just an exploit versus a zero-day exploit.

Two types of techniques to evade detection through obfuscation of a malicious payload are metamorphism and polymorphism. "Metamorphism uses instruction replacement, equivalent semantics, instruction reordering, garbage insertion, and/or register renaming to evade signature-based detectors." Malicious payload would be considered metamorphic if the payload was functionally equivalent to its original form but differed in internal structure. This would allow for the possibility of hard-to-detect viruses and worms to be inserted into a payload while giving the appearance of the original form, thus allowing it to be undetected by defense in-depth defenses."Polymorphism usually uses a built-in encoder to encrypt original shell-code and stores the encrypted shell-code and decryption routine in payload" (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011, p. 271). The encryption of the malicious payload allows for the code to proliferate without being detected.

Intrusion detection and intrusion prevention signatures utilize parts of the four defense techniques previously mentioned. These signatures need to have two basic qualities. "First, they should have a high detection rate; i.e., they should not miss real attacks. Second, they should generate few false alarms" (Yegneswaran, Giffin, Barford & Jha, 2005, p. 14).The goal of any techniques used by an organization should be to detect in real time the existence of a zero-day exploit and prevent damage and proliferation of the zero-day exploit.

3. Statistical-based defense technique

Statistical-based techniques for the detection of exploits rely on attack profiles from past exploits that are now publically known. From those known exploits this defense technique adjusts the historical exploit's profile parameters to detect new attacks.

The quality of the detection is directly related to threshold limits set by the vendor or security professional using this technique(Kaur & Singh, 2014). This technique determines what normal activity is and anything outside of normal is blocked or flagged. The longer the system that is utilizing this technique is online, the more accurate the system is at learning or determining what normal is. "Existing techniques in this approach perform static analysis and/or dynamic analysis on the packet payloads to detect the invariant characteristics reflecting semantics of malicious codes (e.g., behavioral characteristics of the decryption routine of a polymorphic worm)" (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011, p. 270). This technique attempts to detect the exploit prior to the execution of the actual code.

This technique may result in a high rate of false positives or false negatives based on the thresholds chosen(Kaur & Singh, 2014).For example, suppose an employee normally receives a paycheck for $500 bi-weekly for a year. On the next payroll he receives a paycheck that is for $490. Depending on the threshold set, this may be an abnormality that is flagged. The difference in payroll amount may be legitimate, such as a $10 flu shot expense applied, or it could be an error made by payroll staff. If it is flagged and the $490 is correct it would be a false positive. If the threshold for abnormalities is set to $20 and it is not flagged when an error actually occurred, it would be considered a false negative. This technique is also known for the potentially high processing overhead limiting its ability for real time detection(Kong, Jhi, Gong, Zhu, Liu & Xi, 2011).

An example that uses a statistical-based technique would be Semantics Aware Statistical (SAS) algorithm. This technique couples semantic analysis (by introducing static analysis) and statistical analysis in signature generation process (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011). The first phase of the SAS algorithm is the semantic-aware signature extraction phase and is followed by the semantic-aware signature matching phase. In Figure 1 the framework of this approach is put into graphical form. The first phase is broken out into modules which are payload extraction, payload disassembly, useful instruction distilling, clustering, and signature generation. The semantic-aware signature matching phase is comprised of two modules, which are payload extraction and signature matching modules. The payload extraction module extracts the payload that possibly implements the malicious intent from a flow, which is a set of packets forming a

message. The signature matching module starts detecting worm packets by matching state-transition-graph (STG) signatures against input packets" (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011, p. 272-273).

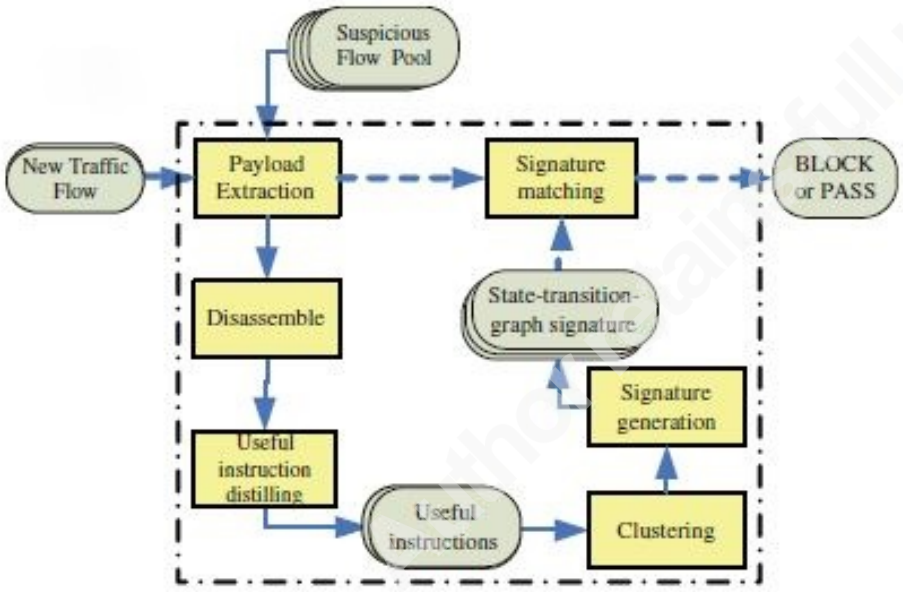

Figure 1 - Semantic-aware Signature Extraction Phase (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011, p. 272).

This particular technique does a few things well. It has the ability to filter noise that is often injected into the packets to "mislead the classifier of the malicious traffic" (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011, p. 269). This allows for cleaner signature generation rather than a learned signature with noise. The STG signature is more complex than previous signatures making it more difficult for attackers to craft packets to circumvent the signature generation process. Since the technique is based on semantic patterns, even if packets are highly modified by attackers, this technique should detect them. The technique also has relatively low overhead which allows it to detect exploits in real time (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011).

Limitations of the SAS algorithm include the inability to handle some state of the art obfuscation techniques, including branch function obfuscation (Kong, Jhi, Gong, Zhu, Liu & Xi, 2011). "Note that a branch function does not return to the instruction following

the call instruction, but instead branches to some other location in the program that depends, in general, on where it was called from" (Linn & Debray, 2003, p. 294). The second limitation of this technique is that by using sophisticated encryption it is possible for attackers to avoid detection.

4. Signature-based defense technique

Signature-based detection is often used by virus software vendors who will compile a library of different malware signatures. They will cross reference these signatures with local files, network files, email or web downloads depending on settings chosen by the user. These libraries are constantly being updated for new signatures that often represent the signatures of new exploited vulnerabilities. The technique is often one step behind a zero-day exploit because this technique requires a signature to be in the signature library for detection. This is the reason virus software vendors are frequently updating their virus definitions.

Signature-based techniques are classified by content-based, semantic-based and vulnerability-based signatures and are somewhat effective against polymorphic worms (Kaur & Singh, 2014, p. 95). The payloads of polymorphic worms change after every attempted infection making them a challenge for security professionals to detect (Mohammed, Chan, Ventura & Pathan, 2013).Their constant change allows them to attack with a signature that is different than their previous attack signature, making them hard to detect and thus allowing them to cause more damage over a longer period of time. Signature-based techniques are used frequently in virus software packages and are often used to defend against malicious payloads from malware to worms.

The first type of signature-based technique mentioned above is content-based. Content-based signatures compare the content of packets with known malicious signatures."Content-based signatures can be classified into the content, the image attributes, which is used as the input for the digital signature algorithm" (Dittmann, Steinmetz & Steinmetz, 1999, p. 210).Content signature-based techniques capture the features specific to worm implementation, thus might not be generic enough and can be evaded by other exploits. Furthermore, various attacks can evade the content-based

signatures by misleading signature generation processes by using crafted packets injection into normal traffic (Kaur & Singh, 2014). Any change in the structure of a malicious packet will often lead to a false negative.

Polygraph is an example of a content signature-based technique that will produce signatures to match and detect polymorphic worms. Newsome, Karp and Song(2005), creators of Polygraph, argue "that it is possible to generate signatures automatically that match the many variants of polymorphic worms, and that offer low false positives and low false negatives" (p. 2). In order for a "real-world exploit to function properly, multiple invariant substrings must often be present in all variants of a payload; these substrings typically correspond to protocol framing, return addresses, and in some cases, poorly obfuscated code" (Newsome, Karp & Song, 2005, p. 1). Polygraph is able to generate signatures based on these invariant substrings.

The second type of signature mentioned above is semantic-based. Semantics is the study of the meaning. The primary goal of semantics is to uncover the meaning of an expression as a whole. "Linguists who study semantics look for general rules that bring out the relationship between form, which is the observed arrangement of words in sentences and meaning" (Thomason, 1996, p.1). Semantic signature-based techniques "are computationally expensive to generate as compared to approaches based on substrings. Moreover, they cannot be implemented in existing Intrusion Detection Systems like Snort" (Kaur & Singh, 2014, p. 95).

An example of a technique that uses semantic analysis is Nemean, "a system for automatic generation of intrusion signatures from honeynet packet traces" (Yegneswaran, Giffin, Barford & Jha, 2005, p. 1). More than two honeypots is called a honeynet and can be implemented as part of a network intrusion detection system. A honeynet can simulate a production environment and is used by security professionals to monitor and log activity of an attacker. "Nemean aims to create signatures that result in lower false-alarm rates by balancing specificity and generality" (Yegneswaran, Giffin, Barford & Jha, 2005, p. 1). They "argue that these capabilities are essential for automatic signature generation systems for the following reasons:

1. Semantics awareness enables signatures to be generated for attacks in which the exploit is a small part of the entire payload.
2. Semantics awareness enables signatures to be generated for multi-step attacks in which the exploit does not occur until the last step.
3. Semantics awareness allows weights to be assigned to different portions of the payload (e.g., timestamps, sequence numbers, or proxy-cache headers) based upon their significance.
4. Semantics awareness helps produce generalized signatures from a small number of input samples.
5. Semantics awareness results in signatures that are easy to understand and validate" (Yegneswaran, Giffin, Barford & Jha, 2005, p. 2)

The two components of the Nemean Architecture are the data abstraction component and the signature generation component. See figure 2.

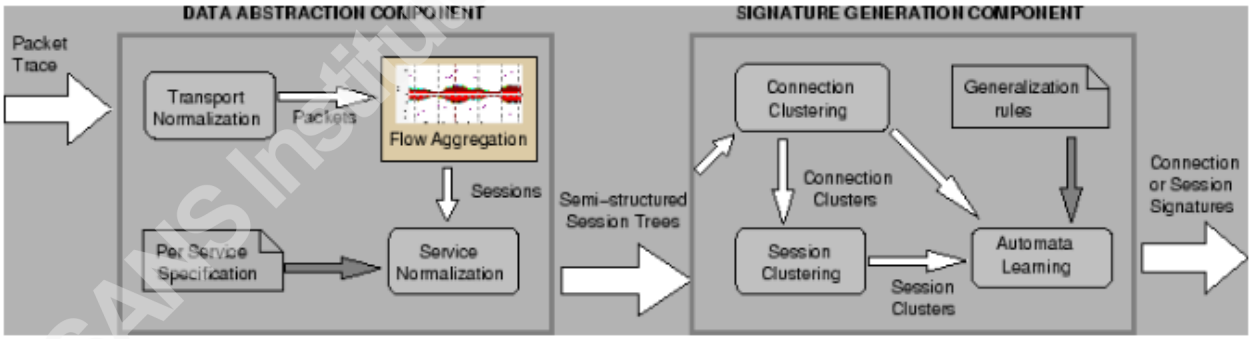

Figure 2: Two components of the Neman Architecture (Yegneswaran, Giffin, Barford & Jha, 2005, p. 2)

The third type of signature mentioned above is vulnerability-based. A vulnerability-based signature has to identify the "vulnerability condition and identify the vulnerability point reachability predicate" (Caballero, Liang, Poosankam & Song, 2009, p. 162). The vulnerability point reachability predicate (VPRP) is a "condition that denotes whether an input message will make the program execution reach the vulnerability point" (Caballero, Liang, Poosankam & Song, 2009, p 162). These types of signatures are based

entirely on the known vulnerability and not on an actual exploit. Since they are based on a known vulnerability they have very few false positives. A weakness of vulnerability-based signatures is the limited library of known vulnerabilities(Kaur & Singh, 2014, p. 95).

A vulnerability-based signature approach designed by Almorsy, Grundy and Ibrahim (2012) is three-fold. "(i) A formal vulnerability definition schema that captures every detail related to a given vulnerability. This helps in every security analysis task, as discussed above; (ii) a formal vulnerability signature specification approach that can capture security vulnerability signatures; and(iii) an extensible vulnerability analysis tool that performs signature-based program analysis. Here, we introduce a static analysis component only"(p. 103). See figure 3.

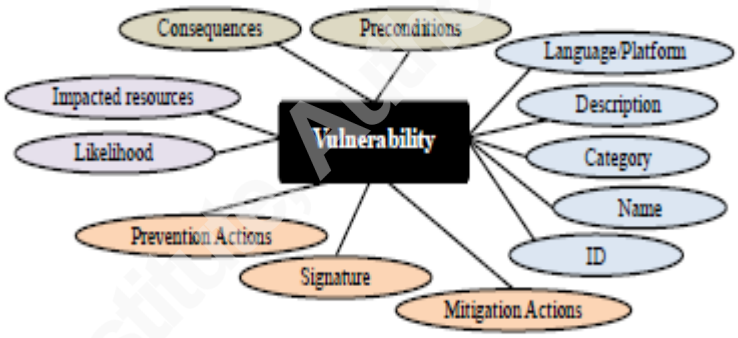

Figure 3: Weakness definition schema (Almorsy, Grundy & Ibrahim, 2012 p. 103)

The key challenge to all signature-based technique defenses is how to accurately and automatically generate signatures in real time that produce low false negatives and low false positives. Once this challenge is met, it can then be introduced in off-the-shelf products which will result in more accurate denials of zero-day exploits (Caballero, Liang, Poosankam & Song, 2009).

5. Behavior-based defense technique

The activity of a program can be viewed as malicious or benign based on the requirements of the code. "Behavior-based techniques look for the essential characteristics of worms which do not require the examination of payload byte patterns"

(Kaur & Singh, 2014, p. 1). The goal of such techniques is to predict the future behavior of a web server, server or victim machine in order to deny any behaviors that are not expected. Those behaviors are learned by the current and past interactions with the web server, server or victim machine (Alosefer & Rana, 2011). This technique relies on the ability to predict the flow of network traffic.

Alosefer and Rana (2011) "propose a malicious activity detection method using Hidden Markov Models (HMM) alongside a client honeypot system" (p. 31). HMM can be a very complex concept to understand. In a Markov model, like a Markov chain, the state is visible to the end user. In a HMM, the state is not visible but the outcome is. Figure 4 uses an example related to weather to illustrate this concept and represents a simple presentation on HMM. The Alosefer and Rana algorithm is able to predict the behavior of a system based on past and current interactions. These behaviors are learned based on the recordings from honeypot systems which are comprised of state machines. Any changes to those state machines are analyzed, and based on those changes, future activity is predicted using the HMM and Baum-Welch algorithm (Alosefer & Rana, 2011).

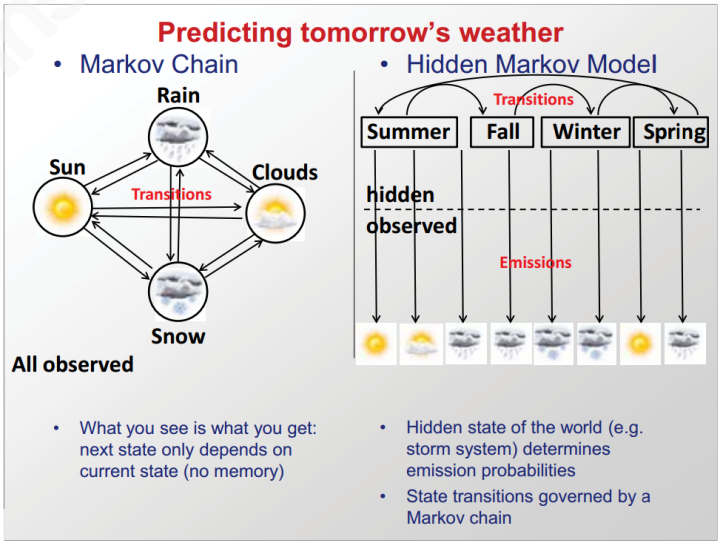

Figure 4: HMM – Simple Explanation (Dernoncourt, 2012)

6. Hybrid-based defense technique

Hybrid-based techniques combine heuristics with various combinations of the three previous techniques which are statistical-based, signature-based, and behavior-based techniques. Using a hybrid model technique will overcome a weakness in any single technique (Kaur & Singh, 2014).

Kaur and Singh used a hybrid technique, Suspicious Traffic Filter (STF), for detecting zero-day polymorphic worms. The benefits of their hybrid technique are four fold:

- Proposal of an efficient technique that offers better sensitivity and specificity by identifying zero-day attacks from data collected automatically on high interaction honeypots.
- Strengthening of the basic existing techniques by combining the advantages of existing techniques and minimizing their disadvantages.
- This technique does not need prior knowledge of zero-day attacks and uses Honeynet as an anomaly detector.
- This technique can detect zero-day attacks in its early phase and can contain the attack before major consequences occur(Kaur & Singh, 2014).

The two techniques Kaur and Singh combined were signature-based and anomaly-based, which fall into the category of behavior-based.

Their technique first tries to detect zero-day polymorphic worms and then tries to quarantine them. “STF observes all network traffic at an edge network and the Internet. The traffic is passed simultaneously to both Honeynet and IDS/IPS (Intrusion Detection System/Intrusion Prevention System) sensors through a port mirroring switch” (Kaur & Singh, 2014, p. 97).

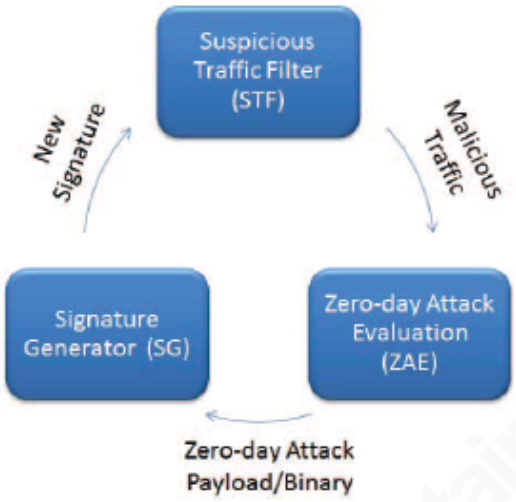

Figure 5: Suspicious Traffic Filter (Kaur & Singh, 2014, p. 97)

Figure5 above "shows the three main components such as Suspicious Traffic Filter (STF), Zero-day Attack Evaluation (ZAE) and Signature Generator (SG). STF is the first defense layer from zero-day attack. ZAE takes input (malicious traffic) from STF to evaluate and analyze captured zero-day attack. SG generates new signature for zero-day attack and updates the signature database in STF. These three main components will work together as interrelated process" (Kaur & Singh, 2014, p. 97).

7. Organizational size relationship to zero-day exploit defense

Two of the main differences and challenges faced by various sized organizations against the defense of zero-day exploits are knowledge and awareness of these exploits, and available and applied resources for their defense. These differences and challenges may be equalized, regardless of an organization's size, by compliance requirements imposed on particular organizations by acts such as Health Insurance Portability and Accountability Act (HIPPA) and Sarbanes-Oxley Act (SOX).

The smaller the organization the more likely the organization has less formalized policies and procedures in regards to security. Often these organizations are unaware of the potential risks from zero-day exploits and therefore do not know the importance of defending against them. Medium to large organizations generally have formalized

policies and procedures and greater knowledge of the risks, and, consequently, are more likely to try to defend against them.

Resources available to an organization have a direct role in its ability to defend against zero-day exploits. Available resources include knowledgeable security personnel, software, and hardware. Large to medium size organizations can usually implement a better defense in depth strategy using various techniques to defend against zero-day exploits.

Andrew Briney and Frank Prince conducted a study in 2002 regarding security and organization size titled "Does Size Matter?" Information technology (IT) hardware, software and concepts have changed greatly since 2002; but, this study's results would be similar if the study was reproduced in 2014 because little has changed in regards to organizational structure. The study surveyed 2,196 IT security practitioners during May and June 2002 and revealed there are distinct security defense patterns based on organization size. The main difference between small and large companies often is that large organizations have better technology to detect problems and know they have been hacked and the small companies continue to operate without awareness of the infiltration.

Small organizations in the study are defined as organizations with 10-100 computers. IT staff usually are performing multiple functions. Of the small organizations surveyed, 50% had a dedicated security staff. Per user costs related to security is the highest of all the organization sizes. If a small organization's culture embraces security and security awareness is in the forefront when designing their network, the organization can be secure. Smaller organizations have an easier time implementing security policies because of their size but usually have fewer security policies. These organizations utilize off-the-shelf security programs and have little to no custom modifications and are likely unaware of how the protection they are implementing is defending their organization.

Medium-size organizations in the study are defined as organizations with 100-1,000 computers. Like smaller companies, IT staff is usually performing multiple functions. Two-thirds of the medium-sized organizations had at least one full time IT staff. Per user costs related to security are less than small organizations. These organizations often have a hard time implementing security policies. Security

effectiveness of medium-sized organizations is based mostly on senior management's focus. Among all of the organization sized groups, this one scored the lowest in making security decisions based on approved policies. These organizations utilize off-the-shelf security programs and have little to no custom modifications. The focus of senior management will determine the depth of knowledge of their IT staff and how they defend their environment.

Large organizations in the study are defined as organizations with more than 1,000 computers. These organizations have multiple security professionals designing, implementing or monitoring their defense in depth strategy. The dollar amount large companies spend on security per user is the lowest of all of the groups although these organizations spend the largest total dollar amounts on security. These organizations function based on written policies and procedures and grant their users only enough security access to perform their job function. Security staff is trained well and know how their defense in depth strategy defends their environment.

8. Conclusion

Most of the defense techniques available to organizations today are available in off-the- shelf hardware and software applications. The methods used by hardware and software applications are usually defined as a hybrid model. In order to best defend against zero-day exploits, an organization needs to understand what defense techniques their defense in depth strategy defends against. The ability for a smaller organization to defend itself versus a large organization is often limited by knowledge of the threat by IT staff and senior management, as well as limited financial resources. The amount of information available to users and management is growing daily. Through organizations like SANS, security journals, and media outlets, organizations can benefit by educating decision makers on zero-day exploit risks and defense approaches so that informed action can be taken to minimize the possible impact in the future.

David Hammarberg, dhammarberg@macpas.com

References

Almorsy, M.; Grundy, J.; Ibrahim, AS., "Supporting automated vulnerability analysis using formalized vulnerability signatures," Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on, vol., no., pp.100,109, 3-7 Sept. 2012

Alosefer, Y.; Rana, O.F., "Predicting client-side attacks via behavior analysis using honeypot data," *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on Next Generation Web Services Practices,* pp.31,36, 19-21 Oct. 2011.

Briney, A., & Prince, F. (2002). Does Size Matter. Information Security, 5(9), 36-39.

Caballero, J., Liang, Z., Poosankam, P., & Song, D. (2009, January). Towards generating high coverage vulnerability-based signatures with protocol-level constraint-guided exploration. In Recent Advances in Intrusion Detection (pp. 161-181). Springer Berlin Heidelberg.

Chen Ting; Zhang Xiaosong; Liu Zhi, "A Hybrid Detection Approach for Zero-Day Polymorphic Shellcodes," *E-Business and Information System Security, 2009. EBISS '09. International Conference on*, vol., no., pp.1, 5, 23-24 May 2009 doi: 10.1109/EBISS.2009.5137874

Colberg, C., Thimborson, C., Low, D.: A taxonomy of obfuscating transformations. In: Technical Report 148, University of Auckland (1997)

Dernoncourt Franck, (November 2012). *What is a Simple Explanation of the Hidden Markov Model Algorithm* Retrieved from http://www.quora.com/What-is-a-simple-explanation-of-the-Hidden-Markov-Model-algorithm.

Dittmann, J.; Steinmetz, A; Steinmetz, R., "Content-based digital signature for motion pictures authentication and content-fragile watermarking," Multimedia Computing and Systems, 1999. IEEE International Conference on, vol.2, no., pp.209, 213 vol.2, Jul 1999

Kaur, R.; Singh, M., "Efficient hybrid technique for detecting zero-day polymorphic worms," *Advance Computing Conference (IACC), 2014 IEEE International* , pp.95,100, 21-22 Feb. 2014.

David Hammarberg, dhammarberg@macpas.com

Kong, D., Jhi, Y., Gong, T., Zhu, S., Liu, P., & Xi, H. (2011). SAS: semantics aware signature generation for polymorphic worm detection. International Journal of Information Security, 10(5), 269-283. doi: 10.1007/s10207-011-0132-7

Leyla Bilge and Tudor Dumitras (ACM Conference on Computer and Communication Security). "Before We Knew it: An Empirical Study of Zero-Day Attacks in the Real World." October 2012.

Mohammed, M.M.Z.E.; Chan, H.A; Ventura, N.; Pathan, A-S.K., "An Automated Signature Generation Method for Zero-Day Polymorphic Worms Based on Multilayer Perceptron Model," Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on , vol., no., pp.450,455, 23-24 Dec. 2013

Newsome, J.; Karp, B.; Song, D., "Polygraph: automatically generating signatures for polymorphic worms," Security and Privacy, 2005 IEEE Symposium on, vol., no., pp.226, 241, 8-11 May 2005

Nzoukou, W.; Lingyu Wang; Jajodia, S.; Singhal, A, "A Unified Framework for Measuring a Network's Mean Time-to-Compromise," *Reliable Distributed Systems (SRDS), 2013 IEEE 32nd International Symposium on Reliable Distributed Systems,* pp.215,224, Sept. 30 2013-Oct. 3 2013.

Symantec Corporation, (April 2014). *Internet security threat report* Retrieved from http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf.

Thomason, R.H. (2012, March 27). *What is Semantics?* Retrieved from http://web.eecs.umich.edu/~rthomaso/documents/general/what-is-semantics.html

Wang Wei; Luo Dai-sheng; Zhang Jianmin, "Detect Polymorphic Worms Based on Semantic Signature and Data-Mining," Communications and Networking in China, 2006. ChinaCom '06. First International Conference on, vol., no., pp.1, 4, 25-27 Oct. 2006

Yu, Y., Wenlong, X., Andong, Q., Ge, Y., & Fuxiang, G. (2012). Hopf Bifurcation in an SEIDQV Worm Propagation Model with Quarantine Strategy. Discrete Dynamics in Nature & Society, 1-18. doi:10.1155/2012/304868

David Hammarberg, dhammarberg@macpas.com

Last Updated: August 22nd, 2018

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Copenhagen August 2018	Copenhagen, DK	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS SEC504 @ Bangalore 2018	Bangalore, IN	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, JP	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, NL	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Wellington 2018	Wellington, NZ	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS MGT516 Beta One 2018	Arlington, VAUS	Sep 04, 2018 - Sep 08, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FLUS	Sep 04, 2018 - Sep 09, 2018	Live Event
Threat Hunting & Incident Response Summit & Training 2018	New Orleans, LAUS	Sep 06, 2018 - Sep 13, 2018	Live Event
SANS Baltimore Fall 2018	Baltimore, MDUS	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS Alaska Summit & Training 2018	Anchorage, AKUS	Sep 10, 2018 - Sep 15, 2018	Live Event
SANS Munich September 2018	Munich, DE	Sep 16, 2018 - Sep 22, 2018	Live Event
SANS London September 2018	London, GB	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NVUS	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS DFIR Prague Summit & Training 2018	Prague, CZ	Oct 01, 2018 - Oct 07, 2018	Live Event
Oil & Gas Cybersecurity Summit & Training 2018	Houston, TXUS	Oct 01, 2018 - Oct 06, 2018	Live Event
SANS Brussels October 2018	Brussels, BE	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Amsterdam October 2018	Amsterdam, NL	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Riyadh October 2018	Riyadh, SA	Oct 13, 2018 - Oct 18, 2018	Live Event
SANS Northern VA Fall- Tysons 2018	Tysons, VAUS	Oct 13, 2018 - Oct 20, 2018	Live Event
SANS October Singapore 2018	Singapore, SG	Oct 15, 2018 - Oct 27, 2018	Live Event
SANS London October 2018	London, GB	Oct 15, 2018 - Oct 20, 2018	Live Event
SANS Denver 2018	Denver, COUS	Oct 15, 2018 - Oct 20, 2018	Live Event
SANS Seattle Fall 2018	Seattle, WAUS	Oct 15, 2018 - Oct 20, 2018	Live Event
Secure DevOps Summit & Training 2018	Denver, COUS	Oct 22, 2018 - Oct 29, 2018	Live Event
SANS Houston 2018	Houston, TXUS	Oct 29, 2018 - Nov 03, 2018	Live Event
SANS Gulf Region 2018	Dubai, AE	Nov 03, 2018 - Nov 15, 2018	Live Event
SANS Sydney 2018	Sydney, AU	Nov 05, 2018 - Nov 17, 2018	Live Event
SANS Dallas Fall 2018	Dallas, TXUS	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS London November 2018	London, GB	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS DFIRCON Miami 2018	Miami, FLUS	Nov 05, 2018 - Nov 10, 2018	Live Event
Pen Test HackFest Summit & Training 2018	Bethesda, MDUS	Nov 12, 2018 - Nov 19, 2018	Live Event
SANS Osaka 2018	Osaka, JP	Nov 12, 2018 - Nov 17, 2018	Live Event
SANS San Francisco Summer 2018	OnlineCAUS	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced