Intrusion Detection: Tools, Techniques and Strategies

Vijay Anand
Industrial Engineering and technology
Southeast Missouri State University
Cape Girardeau, USA 63701
vanand@semo.edu

ABSTRACT

Intrusion detection is an important aspect of modern cyber-enabled infrastructure in identifying threats to digital assets. Intrusion detection encompasses tools, techniques and strategies to recognize evolving threats thereby contributing to a secure and trustworthy computing framework. There are two primary intrusion detection paradigms, signature pattern matching and anomaly detection. The paradigm of signature pattern matching encompasses the identification of known threat sequences of causal events and matching it to incoming events. If the pattern of incoming events matches the signature of an attack there is a positive match which can be labeled for further processing of countermeasures. The paradigm of anomaly detection is based on the premise that an attack signature is unknown. Events can deviate from normal digital behavior or can inadvertently give out information in normal event processing. These stochastic events have to be evaluated by variety of techniques such as artificial intelligence, prediction models etc. before identifying potential threats to the digital assets in a cyber-enabled system. Once a pattern is identified in the evaluation process after excluding false positives and negative this pattern can be classified as a signature pattern. This paper highlights a setup in an educational environment to effectively flag threats to the digital assets in the system using an intrusion detection framework. Intrusion detection framework comes in two primary formats a network intrusion detection system and a host intrusion detection system. In this paper we identify different publicly available tools of intrusion detection and their effectiveness in a test environment. This paper also looks at the mix of tools that can be deployed to effectively flag threats as they evolve. The effect of encryption in such setup and threat identification with encryption is also studied.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – Access controls, Authentication, Cryptographic controls, Information flow controls, Verification.

General Terms

Management, Performance, Reliability, Security, Verification.

SIGUCCS'14, November 2–7, 2014, Salt Lake City, UT, USA.
Copyright 2014 ACM 978-1-4503-2780-0/14/11…$15.00
http://dx.doi.org/10.1145/2661172.2661186

Keywords

Attacks; Virtualized; Intrusion; Pattern; Anomaly; Honeypot; Honeynet; Sanitization

1. INTRODUCTION

The reference to cyber enabled infrastructure implies an infrastructure whose access and control is enabled by computing hardware and software at minimum and may additionally have networking components for remote access and control. As cyber-enabled infrastructure has become ubiquitous, interest in this infrastructure has also increased. Some of this interest is maleficent which has a lot of potential to damage. There is a need to identify such attacks on to system. Attacks that can be staged on a cyber-enabled infrastructure needs to be detected for any corrective solutions. Hence the requirement for intrusion detection is very important. The detection aspect does not necessarily mean prevention, but it triggers countermeasures which may or may not always result in prevention. Attacks are of two kinds, passive and active. A passive attack is one in which the attacker is not controlling the attack outcomes in real time where as an active attack is one in which an attacker is basing actions on the feedback received on outcome of an attack. In this paper we highlight the different types of attacks and their classification, the available techniques to detect these attacks, with an emphasis on the cryptographic aspect of a cyber-enabled infrastructure, available tools for detecting attacks, setup and insight about outcomes.

2. ATTACK CLASSIFICATION

An attack into a cyber-enabled system broadly categorizes the disruption of confidentiality, integrity, authenticity, availability, privacy and value of a digital asset. Digital Assets in this context refers to:

- Digital personal data (Email, Facebook content etc.),
- Computing Systems(Server, PC, Tablets, Smartphones),
- Networking Infrastructure(Wireless(802.11, 3G, LTE), Wired)

Attack types on a cyber-enabled system can be broadly classified into active and passive attacks [1]. This method of classification is chosen based on the adversary type over the details of the attack. Detection systems need to adapt on the type of adversary and detection engines need to quantify behavior for effective attack identifications. Active attacks are attacks where the attacker is molding attacks based on the system response [2]. A cyber enabled system is an automaton whose responses are predetermined by the system designer. An active attacker can look at the system response and intelligently stage attacks to gain access to a digital asset. Detection of such attacks can be difficult. Passive attacks [1][3] on the other hand have no real time intelligence associated with it. The attack can be classified as automaton attacking another automaton. The attack automaton does have some intelligence but it is predetermined by the automaton creator which can be simple or sophisticated. The exploits these attacks leverage can be quantified on software and system vulnerabilities. The exploit classification can be broadly put into:

- Input Based Exploits: these types of exploits includes the set of threats which take advantage of vulnerabilities in input handing and input events of a computing service. The different vulnerabilities are Boundary Condition Error, Input Validation Error, Failure to Handle Exceptional Conditions, Origin Validation Error. These exploits can be controlled through indirectly or directly derived from inputs.
- Environment Based Exploits: This category of exploits are based on threats emanating from vulnerabilities that change the constructs upon which a cyber-enabled system is designed upon. An exploits are based on either changing the configuration or the environment for which the system was designed.
- Design Errors Exploits: This category of attacks is based on threats in multi-process systems where components of a system are designed without proper interaction to avoid exploitation (i.e., design error attacks)...
- Ciphering System Exploits: This type of exploits represents the threats on the Access Validation Errors, which are encountered within a ciphering system framework due to faulty design or implementation.
- Unknown Exploits: This category of exploit, does not fall under any of the broadly recognized exploit types, and requires individual analysis and treatments to understand the exploits.

There have been few studies in categorization [1] [4] [5] of attacks. In this paper we categorize the attacks based on the subversion of the underlying principle of secure design and architecture. These exploits comprise an attack which can be broadly classified as:

Privilege Escalation attacks: This category of attack tries to subvert the principle of least privilege [6] in secure system design. The objective of the privilege escalation is the ability to get access to digital assets beyond the boundaries of a given privilege.

Access Disruption attacks: These attacks are the most common ones manifesting more broadly as Denial of Service [7] attacks.

Depth of Defense [8] attacks: These attacks are based on the premise that a pivot which does not belong to important services in a cyber-infrastructure is used to gain access to important services.

Confidentiality attacks: These attacks are based on weak ciphers or vulnerable crypto systems that are used in a cyber-infrastructure.

Integrity Attacks: This is similar to confidentiality attacks in the sense that vulnerable algorithms are used in cyber infrastructure.

3. The Setup

A typical laboratory environment for computing studies has a set of licensed computers and some servers that are accessed through the equipment. A high level overview of a laboratory setup is shown below. An important thing to note that with the advent of Bring Your Own Device (BYOD) policy the laboratory environment is much more diverse. The only similarity that is expected from all the devices is that of software running in the systems that is used to access licensed software in the servers in the laboratory framework. In an old configuration the problem lies how many intrusion detection systems need to be put. A laboratory with “X” number of computers which are homogenous in their build would potentially require a single intrusion detection system. The other intrusion detection systems have to match the server abilities. Depending on the types of the software the server or a collection of servers have the number of Intrusion Detection System (IDS) each fine-tuned for a specific datatype has to be chosen. This kind of network intrusion detection if there is no end to end encryption. That does not solve issues if the attack happens after the decryption of data. In such a case there is a need for Host Intrusion Detection System (HIDS) [9] to effectively identify the types of intrusions that are encountered in a laboratory setup. The issue after this is that of storing the logging information and such logging typically should be on a Network Area Storage (NAS), away from the devices to collect and analyses. Different tools can be used for post analysis of this data like Suricata.

As shown in the diagram below there is a firewall associated with this network along with the Intrusion Detection System. The firewall is a layer 2/3 device on the network which can discriminate traffic between endpoints and apply some policy based rules within end points. The firewall does not have the capability to inspect the content of a network connection though where the challenge lies in attempt to identify an intrusion.

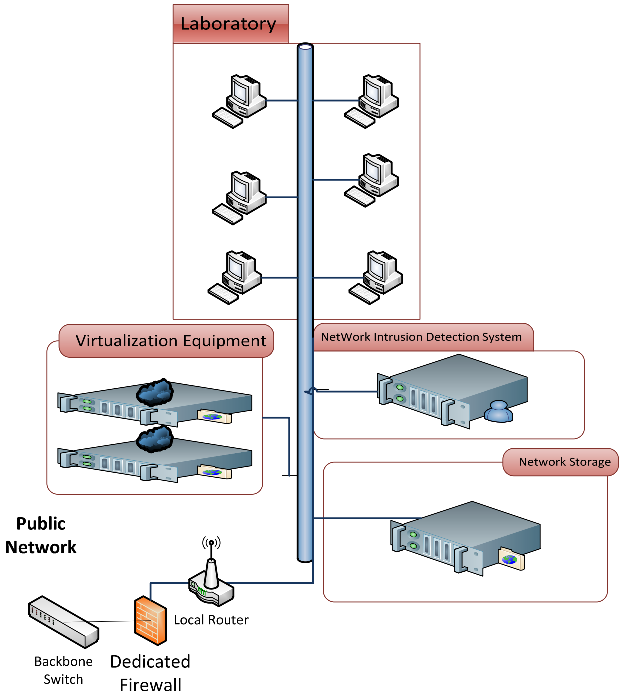

Figure 1: A laboratory setup with Intrusion Detection System

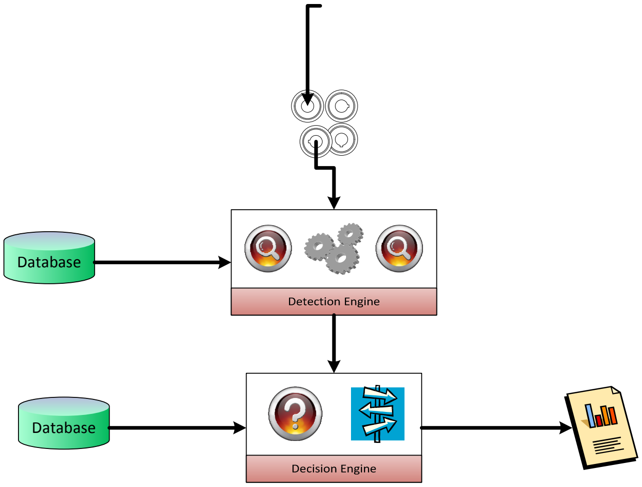

Figure 2: Components of an Intrusion Detection System

4. Intrusion Detection System

There are a few different entities in a cyber-infrastructure that are installed to protect digital assets. The first one is firewall at the network perimeter. As highlighted in Figure 1 a firewall is layer2/3 device [10] that only look at network end points to make decisions. The firewall has no visibility into the data in network. A firewall always sits on the path of the traffic and can potentially be a bottleneck in case the firewall cannot take decisions in real time. Another important element in digital asset protection is that of an anti-malware [11] software which looks at applications and flags applications with known patterns of malware signatures. The anti-malware software is very operating system specific [12][13] and typically looks at stored procedures in an application to make a decision. It is pretty effective on non-active attacks where the malware is resident on a system and the system then analyzes the application to sort a legitimate application from malware. The anti-malware software typically operates at an application layer and hence has visibility to content of network endpoint. But an anti-malware software is not capable of analyzing or detective an active attack. This is where an Intrusion Detection System [14] plays a major role in identifying an attack at the application level. A typical Intrusion Detection System has the components as shown in figure 2. All application data from the content of the end point of networked element is passed through a detection engine. The detection engine then looks at stored patterns of intrusion signatures for live and non-active attacks and then logs the information. A thing to be noted is that the intrusion detection system does not have an actionable outcome [14]. The actionable outcome is left to the system operator to look at log files for analysis and then modify operating system constructs' and firewall policies accordingly. If an intrusion detection system actively modifies the operating system constructs or the firewall constructs such a system is referred to as an intrusion prevention system (IPS) [16]. Intrusion Detection Systems are broadly classified with respect to placement within a cyber-infrastructure as Host based or Network based [13][14][15]. A network based intrusion detection system is an independent installation of other elements of a cyber-infrastructure that passively monitors the network behavior [16]. IDS can be an active device but since it does not have any preventive role it is typically installed as a passive device. As a passive device the IDS profiles all the traffic entering within the perimeter of a network and logs interesting traffic that has a match within its signature database. A Host based IDS[17] on the other hand is an application software that is installed on an operating system which profiles traffic at the host level and logs interesting traffic. Host based traffic typically would execute as an independent process within a system. Some IDS could be executed in both Network and Host mode and these are generally referred to as a Hybrid IDS [18]. The core of IDS as shown in the diagram is that of the detection engine. An important categorization of IDS based on detection engine provides another nuance for choosing IDS for an appropriate application. The detection engine if is based on pattern detection of known attacks signatures then the IDS is referred to as a signature based IDS system. Signature based IDS [19] are based on matching "pattern" for "known patterns" [20] of activity detrimental to elements in the infrastructure. The Benefits of a signature based IDS are:

- Low alarm rates: Effectively the IDS detection engine has to look up a list of known signatures of attacks and log it or report into the monitoring system if a match is identified.
- Signature based IDS are very accurate.
- Speed: Signature based IDS can be relatively fast since the pattern match is based on what traffic is observed and what signature is stored as an attack signature. Based on the percent match attack quantification is done.

A few pitfalls of a signature based IDS system are:

- Signature of an attack is causal to knowledge of an attack. Hence if new attacks are formulated signature based IDS have no measureable benefit. The quality of IDS is as strong as the signatures from which pattern matching can be done. It therefore requires constant updating of signatures to account for new signatures.
- One of the biggest drawbacks of a signature based IDS on the network are by fragmenting of a packet. If an attack deliberately fragments it packets to deliver an attack attacks can be masked.
- Update of a signature is required on a constant basis and the signature dataset has to update before assuring security within a network.

The most common open source IDS Snort [16] primarily uses the signature based detection and the financial structure of Snort type IDS is based on subscription to getting the latters patterns of attack signature.

Another kind of IDS is based on the premise of anomaly detection [21] wherein the IDS looks for traffics patterns and tries to profile traffic based on mathematical or intelligence based computations. The idea is identify anomalies in traffic patterns and if there is an anomaly detected that traffic set can thereafter be analyzed further to identify further issues. Anomalies in time series data are data points that significantly deviate from the normal pattern of the data sequence. Time series is a sequence of data points, measured typically at successive times, spaced at (often uniform) time intervals. A typical way of using anomaly detection is to learn a model of normal behavior

- Using supervised(**e.g., classification**) or unsupervised method(e.g. Clustering, probabilistic models with latent variables such as Hidden Markovian Models)

Based on this model learnt from a suspicion score is constructed

- function of observed data (e.g., likelihood ratio/ Bayes factor)

- captures the deviation of observed data from normal model
- raise flag if the score exceeds a threshold

This kind of attack detection engine [22] is pretty good to identify new attacks into a system. As a side effect of this technique of intrusion detection the IDS does a thorough screening of all traffic. The downside of this detection engine is that of false positives and false negative in the detection mechanism. It can be very resource exhaustive for capturing all traffic for analysis and performance may not be real time. Of the open source IDS systems "Bro" [23] is the one that is known to an anomaly engine extensively. Once a pattern has been recognizes from anomaly detection that particular pattern can then become a signature.

An intrusion detection engine cannot detect all the attacks that are encounter in a cyber-infrastructure. To maximize the number of intrusions detected in a cyber-infrastructure it is important to having multiple IDS with competing technologies executing simultaneously thereby improving the probability of intrusion detection.

The next aspect for IDS is to creating a testing framework [24] so as to identify that a particular IDS is properly setup. One of the ways to do it is having an attack framework for testing purposes. An important aspect of commissioning and IDS is to test it thoroughly and this attack framework becomes an integral part. It is also recommended that the testing is done periodically after commissioning. This attack framework is based on penetration testing tools known. Since penetration testing tool's attack set changes with time the testing periodically aspect of the IDS installation is a part of due diligence effort on part of security practitioners. An important aspect in the testing time is to have the testing framework be isolated from the real network. Isolation is important since some of the attack vectors have a potential to cross the plane of test systems.

The final aspect of the IDS setup is to identify attackers by some network identifiable features. One way to attract attacks onto a cyber-infrastructure is to have potentially vulnerable software which can be monitored. This can be created by creation of honey nets and honeypots

- Honeypot [25] is a computing construct made up of special software applications that can be easily compromised by a cyber-attack.
- Honeynet [26] is a networking construct whose purpose is to attack attackers and can be easily compromised.

Any traffic that enters and exits through the honeynet/honeypot constructs is deemed as suspicious and needs to be carefully monitored. If a successful attack is constructed further attacks on the computing resources can be launched.

The monitoring server in this setup formulates the instantiation of honeypots and intrusion detection system [27] [28] which allows behavior monitoring for risk analysis. The question which arises in this situation is what types of honeypots need to be setup. Most of the classroom operating system is standard so at least one honeypot with a classroom image has to be setup. One important thing in this setup is that connectivity from this honeypot to any other system is restricted though the firewall. The only element this honeypot can access is the firewall in a cyber-infrastructure. There are certain services that need to be enabled in for students to complete their assignments. The other honeypots should be decided based on the software services that are expected to turn on to complete assignments for a given class. This can vary depending on the scope of the class. Any service that is turned on the network requires a honeypot service equivalent on an operating system. If services are hosted on a Linux machine then a Linux honeypot needs to be established and if services are hosted on a Windows device then a Windows honeypot needs to be established.

Another important aspect of this design is that of cryptographic data that is encountered by endpoints crossing the network. Most installations are networking based IDS. As discussed earlier most NIDS don't have visibility to encrypted traffic hence the challenge to incorporate encrypted traffic is critical in understanding of an intrusion. One of the mechanisms of identifying intrusions in end systems is utilizing host intrusion detection (HIDS) systems. Since the traffic is already decrypted on the host machine the host IDS can detect encrypted data that is missed by the network IDS. This implies that all images that are installed in the school devices require installing a host IDS whole logs are collected for analysis and detection.

The setup that was used in this exercise was having an anomaly detection engine IDS, having a pattern detection IDS, a commercial IDS, a honeypot setup, a testing framework with attack systems.

5. CONCLUSIONS

In this paper the need for IDS and the different kinds of IDS are highlighted. A setup is shown in allowing detection of variety of intrusions into a system including cryptographic type intrusions.

6. REFERENCES

[1] Amiel, Frederic, et al. "Passive and active combined attacks: Combining fault attacks and side channel analysis." Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on. IEEE, 2007.

[2] Serjantov, Andrei, Roger Dingledine, and Paul Syverson. "From a trickle to a flood: Active attacks on several mix types." Information Hiding. Springer Berlin Heidelberg, 2003.

[3] Stefano Zanero. 2009. Wireless Malware Propagation: A Reality Check. *IEEE Security and Privacy* 7, 5 (September 2009), 70-74. DOI=10.1109/MSP.2009.142 http://dx.doi.org/10.1109/MSP.2009.142

[4] Gao, Zhiqiang, and Nirwan Ansari. "Tracing cyber attacks from the practical perspective." Communications Magazine, IEEE 43.5 (2005): 123-131.

[5] Bass, Tim. "Intrusion detection systems and multisensor data fusion." Communications of the ACM 43.4 (2000): 99-105.

[6] Schneider, Fred B. "Least privilege and more." Computer Systems. Springer New York, 2004. 253-258.

[7] Zargar, Saman Taghavi, James Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." Communications Surveys & Tutorials, IEEE 15.4 (2013): 2046-2069.

[8] Rocha, Francisco, Thomas Gross, and Aad van Moorsel. "Defense-in-depth against malicious insiders in the cloud." Cloud Engineering (IC2E), 2013 IEEE International Conference on. IEEE, 2013.

[9] Jacob Zimmerman, Ludovic M, Christophe Bidan, "Experimenting with a Policy-Based HIDS Based on an Information Flow Control Model," Computer Security Applications Conference, Annual, p. 364, 19th Annual Computer Security Applications Conference (ACSAC '03), 2003

[10] Achi, H., A. Hellany, and M. Nagrial. "Network security approach for digital forensics analysis." Computer Engineering & Systems, 2008. ICCES 2008. International Conference on. IEEE, 2008.

[11] Sikorski, Michael, and Andrew Honig. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. No Starch Press, 2012.

[12] Shari Lawrence Pfleeger "Anatomy of an Intrusion" IT Pro, 2010

[13] Raheem A. Beyah, *Michael C. Holloway, and John A. Copeland "Invisible Trojan: An Architecture, Implementation and Detection Method"

[14] Mell, R. B. A. P. "Intrusion detection systems." National Institute of Standards and Technology (NIST), Special Publication 51 (2001).

[15] Zhang, Xinyou, Chengzhong Li, and Wenbin Zheng. "Intrusion prevention system design." Computer and Information Technology, International Conference on. IEEE Computer Society, 2004.

[16] Roesch, Martin. "Snort: Lightweight Intrusion Detection for Networks." LISA. Vol. 99. 1999.

[17] Wagner, David, and Paolo Soto. "Mimicry attacks on host-based intrusion detection systems." Proceedings of the 9th ACM Conference on Computer and Communications Security. ACM, 2002.

[18] Aydın, M. Ali, A. Halim Zaim, and K. Gökhan Ceylan. "A hybrid intrusion detection system design for computer network security." *Computers & Electrical Engineering* 35.3 (2009): 517-526.

[19] Kumar, Sandeep, and Eugene H. Spafford. "A pattern matching model for misuse intrusion detection." (1994).

[20] Lee, Sin Yeung, Wai Lup Low, and Pei Yuen Wong. "Learning fingerprints for a database intrusion detection system." Computer Security—ESORICS 2002. Springer Berlin Heidelberg, 2002. 264-279.

[21] Lazarevic, Aleksandar, et al. "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection." *SDM*. 2003.

[22] Depren, Ozgur, et al. "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks." *Expert systems with Applications* 29.4 (2005): 713-722.

[23] The Bro Network Security Monitor http://www.bro.org/

[24] McHugh, John. "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory." ACM transactions on Information and system Security 3.4 (2000): 262-294.

[25] Provos, Niels. "A Virtual Honeypot Framework." *USENIX Security Symposium*. Vol. 173. 2004.

[26] McCarty, Bill. "The honeynet arms race." *Security & Privacy, IEEE* 1.6 (2003): 79-82.

[27] Thonnard, Olivier, and Marc Dacier. "A framework for attack patterns' discovery in honeynet data." *digital investigation* 5 (2008): S128-S139.

[28] Lee, Wenke, and Salvatore J. Stolfo. "A framework for constructing features and models for intrusion detection systems." *ACM transactions on Information and system security (TiSSEC)* 3.4 (2000): 227-261.