ATTACKING THE IDS LEARNING PROCESSES

Tomáš Pevný, Martin Komoň[*]

Czech Technical University in Prague
Technická 2, 166 27, Prague 6
Czech Republic

Martin Rehák[†]

Cognitive Security s.r.o.
Dřevná 382/2, 120 00, Prague 2
Czech Republic

ABSTRACT

We study the problem of directed attacks on the learning process of an anomaly-based Intrusion Detection System (IDS). We assume that the attack is performed by a knowledgeable attacker with an access to system's inputs, outputs, and all internal states. The attacker uses his knowledge of the IDS (implemented as an ensemble of anomaly detection algorithms) and its internal states to design the strongest undetectable attack of a particular type. We have experimented with different attacks against several anomaly detection algorithms individually, and against their combination. We show that while the individual anomaly detection algorithms can be easily avoided by the worst-case attacker that we assume, it is nearly impossible to avoid them simultaneously. These results were achieved during the experiments performed on university network traffic and are consistent with theoretical hypothesis grounded in steganalysis and watermarking.

1. INTRODUCTION

Network intrusion detection systems (NIDS) are becoming a standard part of security measures protecting enterprise computer networks. NIDS are usually deployed on the perimeter of the protected network, but their role is not limited to the detection of attacks from the outside, as they are also used to scrutize traffic generated within the protected network in order to identify hosts infected by malware [1], to detect information ex-filtration [2], and other types of unwanted traffic (p2p networks, skype, etc.).

Most frequently deployed NIDS, such as SNORT [3] or BRO [4], rely on the signature matching mechanism, where payload of packets is inspected and matched against signatures of known malware and other threats. Despite their wide spread use, they posses many undesirable properties, some of them being the need of keeping database of signatures up to date (implying the inability to detect zero day attacks), inability to scrutinize encrypted traffic, and high computational demands. To alleviate these limitations, a lot of research efforts is devoted to intrusion detection systems based on the anomaly detection paradigm.

Anomaly detection based NIDS [5] relies on the assumption that the attack traffic has different statistics from the benign one. The usual approach to detect attack traffic is to identify outliers with respect to the model of the whole traffic, which is assumed to be mostly benign. The model of whole traffic is usually built online. On one hand, this choice enables adaptation to changes, as it is assumed the trafic to be non-stationary and network-specific. On the other hand, the adaptivity undermines the security of NIDS, since the knowledgeable adversary can manipulate the model, such that the attack traffic will be classified as benign [6, 7].

Attacks on intrusion detection systems are not particularly novel when considered from broader perspective. Throughout the history, countermeasures to and attacks on radar, sonar, IFF systems and related technologies were deployed very early after their introduction. This is also true for the NIDS [8, 7]. However, attacks discussed here stand out by being directed against the detection algorithm itself, rather than against sensors. Attacker uses its knowledge of detection algorithms to probe and progressively mislead the detection and pattern recognition algorithms, so that they are blinded. This category of attacks, based on the Adversarial Machine Learning (AML) will become prominent with the increased use of data mining, machine learning, and AI in general for broader security practice.

In this paper, we concentrate on increasingly important case of attacks on systems, where a set of diverse anomaly detectors works as an ensemble, effectively constituting a single joint detector [9]. This configuration prompts two key questions related to the AML problem:

- Can simple detectors be individually attacked in order to render them blind w.r.t. detection of particular attack?

[*] The work of T. Pevný was supported by the Grant Agency of Czech Republic under the project P103/12/P514.

[†] This material is based upon work supported by the ITC-A of the US Army under Contract W911NF-12-1-0028 and by ONR Global under Contract N62909-12-1-7019. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Government. Also supported by Czech Ministry of Interior grant number VG20122014079.

- Is the ensemble of detectors more or less vulnerable to such attack?

Specifically, the goal is to perform an undetected attack, since as will be argued later, this is a key step in shifting the NIDS' state towards acceptance of large-scale attacks. The investigation is performed under the worst-case scenario for the defender (NIDS), as the attacker has full access to all internal states of the system. In the categorisation introduced by Barreno [6], this is an exploratory, targeted attack on integrity of the detection process.

2. HIDING THE ELEPHANT

Ptacek et al. [8] has identified two, complementary strategies to evade the detection. The *evasion* strategy decreases the intensity of the attack or modifies it such that it will be accepted by the NIDS, while the *insertion* strategy adds supplemental traffic to mask the actual traffic caused by the attack.

The first attacker's strategy is to **lower the strength** of the attack[1] and spread it over longer period of time. The rationale behind is that by making the strength of the attack very small, the proportion of the attack traffic with respect to the total will be low as well. This will make it difficult to be separated from the benign traffic, as it would not be clear, if it is a noise or a signal. Its drawback is that the strength of the attack can be so low that the possible reward for the attacker would not be interesting anymore, and he will look for other target. An example is brute-force cracking of a SSH password, where having only couple trials per minute is practically useless.

Notice that the successful evasion strategy is a key requirement for modification of NIDS' internal states to accept large attack. To shift the internal state to this point, the attacker starts with a small attack and gradually increases its intensity, such that it is just below the detection threshold. By doing so, he hope to reach point, where his large attack is not detected [6]. This strategy is used in experiments described in Section 3.1.

This idea has been experimentally verified by Rubinstein et al. [7], who has targeted the detector of Lakhina et al. [10]. A simpler approach has been investigated by Newsome et al. [11], who proposed to shift just the detection threshold toward accepting the malicious traffic by increasing false positive rate (red-herrink attack). Both works [7] and [11] exploit the fact that targeted systems had distinct training phase (during which the model of the traffic is inferred) and detection phase (during which the system is actually used to detect the attacks). We do not believe these conditions to be realistic, since due to the non-stationarity of the traffic, models need to be constantly updated (learned). Moreover, algorithms in both approaches were allowed to raise as many false alarms, as possible, which we again do not consider to be realistic.

In the second strategy, the attacker **generates additional traffic** not directly related to the attack. Its purpose is to conceal the actual attack, such that the overall statistics of attacker's traffic observed by the detection system look innocuous.

Although the insertion strategy is interesting, we do not believe to be useful in practice. The amount of the additional traffic needed to conceal the attack traffic might be so large that the attacker might not be able to generate it with his limited resources, or it will be easily detectable by detectors monitoring volume of the traffic [10, 12]. Moreover, the additional traffic might disturb some network statistics the attacker is not aware off [2], which can make his activities easily detectable.

Notice here the similarity to steganography, where communicating parties try to hide the secret message into innocuous looking objects (e.g. digital image). In the steganography domain it has been already many times experimentally verified that making more changes, which corresponds to the insertion strategy, increases the probability of being detected. Due to above arguments, we investigate the evasion strategy only, and left the other for a future work. We believe it to be more important, as the attacker has higher probability of being successful with less resources.

3. EXPERIMENTS

3.1. Experimental details

In this section, we evaluate chances of the attacker to successfully plant an evasion attack. We emphasise that we simulate the worst case scenario for the defender, where the attacker has a *full access* to all internal states of the NIDS. We assume the attacker to have limited resources in the sense that the attack is performed only from one subnet (source IPs used by the attacker differ only in the last octet).

To simplify the implementation and speed-up the experiments, all attacks were simulated by generating the attack traffic inside the attacked system. The attacked ensemble of intrusion detection algorithms consisted from a following set of detectors: volume and entropy detectors of Lakhina et al, [10] (Lakhina Volume and Lakhina Entropy), MINDS [12], detectors of XU et al. [13] (Xu sIP and Xu dIP), and the scan detector presented in [14] (TAPS). Since all detectors were initially designed for a backbone traffic, we used their adaptations to enterprise-level networks, described in [15]. Due to the space limit, we cannot describe algorithms in detail here, thus we refer the reader to the corresponding publications.

The detection algorithms processes the traffic in 5-minute long time windows. Due to the continuity, attack flows to be used in the time window $t + 1$ needs to be prepared in

[1] Under the term "strength of the attack", we understand the bandwidth occupied by the attack traffic. The bandwidth can be measured for example by bits / packets / flows per second.

[2] This is actually inconsistent with our assumptions, but yet worth to note

the time window t. They are created such that they are not detected by the detectors at the time window t (the level of anomaly is below threshold $\alpha = 0.05$). For the attacker this means that he cannot be absolutely certain that the attack will be undetected, since internal states and the background traffic have changed. But due to the temporal correlation, attacker's chances of success are very high.

Our experiments used three simple and common types of attack: horizontal and vertical scans, and the brute force cracking of SSH password. We chose them due to their ubiquity and different characteristics. The *horizontal scan* have wide range of destination IPs, usually one destination port, and low number of bytes and packet counts per flow. The *vertical scan* usually have one destination IP, many destination ports, and low byte and packet counts per flow. The *Brute force cracking of SSH password* has one destination IP and port, and high number of packets.

The characteristics not specified for a given type of the attack are free and the algorithm generating attack flows manipulates them in order to make the attack traffic undetectable. The algorithm also verifies that the attack does not lose its main characteristics (e.g. brute-force cracking of SSH password whose destination port range needs to be increased from one to a thousand is not a SSH cracking any more).

The attack traffic is generated as follows. (a) Create the initial attack traffic according to the specification. (b) If the level of anomaly, asserted by modified detector(s), exceeds threshold $\alpha = 0.05$ on scale $[0, 1]$, modify the traffic. The detectors report, why does it happen and recommend the modification (e.g. to increase / to decrease entropy of source ports, to decrease number of flows from / to IP address, etc.). Repeat the step. (c) If no detector raises alarm, the generation is finished and the traffic is mixed with the background traffic.

The algorithm iterates unless the attack is not detected, or the loop exceeds certain number of iterations, or the attack has lost its properties. In the last two cases, it is assumed that the attack cannot be implemented at this time.

The initial strength of the attack is always set to twice the strength of the attack from the previous time window. If the attack in the previous time window was not successful, the strength is set to 150 flows per five-minute long time window. This simple strategy enables (*i*) to reach the maximum strength at exponential rate, and (*ii*) to simulate the shifting of NIDS' internal state toward accepting stronger attacks, as we continuously try to increase the attack strength.

The experiments used background traffic captured from the university network with approximately 25 000 flows per five-minute time window spanning 24 hours. This allows us to observe the effect of working hours and nighttime. The attacks were initiated after 50 minute long warm-up period allowing detectors to reach their working conditions. We emphasise that we do not need the background traffic to be labeled, as our focus is the detection of simulated attack flows, rather than the background flows.

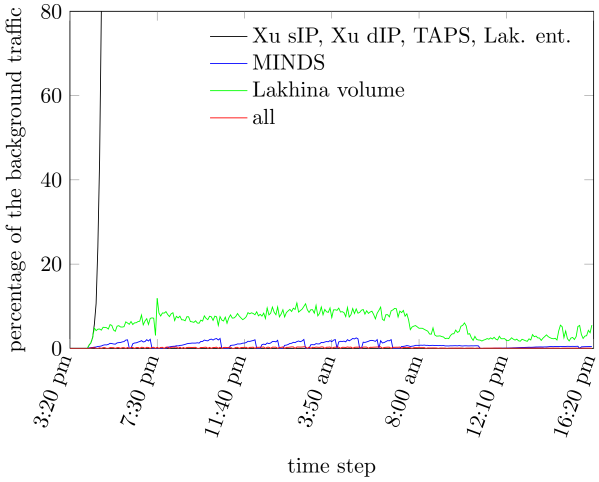

Fig. 1. Evolution of the strength of the brute-force cracking of SSH password targeted to bypass individual detectors, and all detectors collaborating together (label "all"). The strength of the attack is measured as a percentage of total number of flows in background traffic.

3.2. Experimental results

The results of our experiments are summarised in Table 1. We can see that outcomes on different attacks are very similar. Our algorithm was able to create attacks of any strength against individual detectors based on entropies (Xu sIP, Xu dIP, and Lakhnina Entropy). This is an expected result, since these detectors do not take the volume of the traffic into the account, as their rely entirely on the distribution of its flows. Bypassing TAPS detector is very easy, since this detector only scrutinise flows with one packet, hence the algorithm was able to immediately devise attack of any strength.

Contrary, detectors modelling the volume of the traffic (Lakhina Volume, MINDS) proved to be very limiting. Although it was possible to bypass them, the strength of the attack was very limited. It is interesting to observe that the MINDS detector, which is a way simpler than Lakhina Volume, proved to be more efficient in limiting the attack.

Bypassing all detectors simultaneously proved to be almost impossible. In fact, the algorithm has failed except the SSH password cracking, where the strength of the attack was 0.14% of the background traffic. This corresponds to rate of 7 passwords tried per minute, which renders the cracking useless.

Figure 1 shows the evolution of the strength of the brute-force cracking of SSH password (graphs for other attacks are virtually the same, hence they are omitted to save the space). We can observe that strength of attacks against Xu sIP, Xu dIP, Lakhina Entropy, and TAPS detectors exponentially grows to infinity. From this reason, experiments with these detectors were stopped after processing 20 five-minute long snapshots

Detector	horizontal scan		vertical scan		SSH bruteforce	
	strength	success	strength	success	strength	success
Lakhina Volume	13.7%	100%	6.23%	100%	5.89%	100%
Lakhina Entropy	$+\infty$	100%	$+\infty$	100%	$+\infty$	100%
MINDS	2.06%	92.36%	1.09%	93.05%	0.85%	85.06%
TAPS3D	$+\infty$	100%	$+\infty$	100%	$+\infty$	100%
Xu sIP	$+\infty$	100%	$+\infty$	100%	$+\infty$	100%
Xu dIP	$+\infty$	100%	$+\infty$	100%	$+\infty$	100%
All	0	0%	0	0%	0.14%	79.16%

Table 1. The rows shows statistics of attacks designed to bypass individual detectors. The last row ”All” shows the same when bypassing all detectors simultaneously. Columns captioned ”strength” shows the average strength of the attack expressed in percents of flows of background traffic. $+\infty$ means that the algorithm was able to find attack of any strength. Columns captioned ”success” shows the rate, at which the attack was undetected.

(10 snapshots for warm-up and 10 for generating the attack). This is on par with above elaborations, as the first three detectors use entropy measures to assess the level of anomalousness.

Contrary, attacks against Lakhina Volume and MINDS detectors quickly reach their maximum strength and stays at the similar level. It is interesting that the attack against MINDS has its maximum strength during night hours, and from approximately 7:00am (people arriving to university), the strength declines. These observations contradict our initial assumption that attacks are easier to detect when the traffic is low. Our explanation of this phenomenon is that during business hours, there is an additional traffic interfering with the attack. The MINDS detector checks that the absolute number of flows does not exceed the threshold. Thus, if we sum background and attack traffic together, the space left for the attack is smaller during day hours. The same holds for the Lakhina Volume detector.

4. CONCLUSION AND FUTURE WORK

This paper aimed to practically investigate, if a knowledgeable attacker can avoid detection of real NIDS implemented as an ensemble of simple anomaly detectors. We have focused on the evasion strategy of the attacker, because it is simpler, more likely to succeed, and it is an essential component of the strategy, where the attacker tries to modify internal states of well designed NIDS towards accepting large-scale attacks. Experiments assumed the worst-case scenario for the defender (NIDS), where the attacker has full access to all internal states of the NIDS.

The evasion strategy was implemented against the subset of six diverse anomaly detectors deployed in one of the layers of commercially available CAMNEP system. Our results show that while it is possible to bypass individual detectors, it is nearly impossible to do so when detectors are used as an ensemble. This implies that NIDS implemented as an ensemble of diverse detectors is very robust against the modification of its internal state.

This is important for the design of future NIDS. We believe that (*i*) they should by implemented by a mix of detectors, some of them being adaptive, and some of them being static (the latter can prevent the detector's manipulation despite their low detection accuracy); (*ii*) It is important to use detectors modelling the volume of the traffic, as they can fairly limit the strength of the attack.

Strategically, our results imply that well-designed ensemble detectors are currently safe from adversary's manipulation, provided that: (*i*) the attacker is not able to influence the majority of the traffic in the monitored network; (*ii*) the ensemble contains detectors that are able to detect unusually increased traffic volumes required for causative attacks. Elaborating on (*ii*), we argue that detection algorithms mutually protect each other, as the detection capabilities of other detectors limit the volume and properties of the traffic that the attacker can use to attack the target detector. This makes the causative integrity attacks suboptimal, as they would require more traffic than purely exploratory attacks.

We believe that in the long term, the adversarial machine learning approaches will rely on techniques from game theory and game playing fields in order to (*i*) strategically select detectors in the ensembles and (*ii*) to perform the fusion of information within the ensemble.

5. REFERENCES

[1] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting botnet command and control channels in network traffic,” in *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, February 2008.

[2] K. Borders and A. Prakash, “Quantifying information leaks in outbound web traffic,” in *Security and Privacy,*

2009 30th IEEE Symposium on. May 2009, pp. 129–140, IEEE.

[3] Sourcefire, Inc., "SNORT – Intrusion Prevention System," `http://www.snort.org/`, 2011, Accessed in June 2011.

[4] "The Bro Network Security Monitor," `http://bro-ids.org/`, 2011, Accessed in June 2011.

[5] D. Denning, "An intrusion detection model," in *IEEE Security and Privacy*, 1986.

[6] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?," in *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, New York, NY, USA, 2006, pp. 16–25, ACM.

[7] B.I.P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.H. Lau, S. Rao, N. Taft, and J. D. Tygar, "ANTIDOTE: understanding and defending against poisoning of anomaly detectors," *Internet Measurement Conference*, pp. 1–14, 2009.

[8] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," Tech. Rep., Secure Networks, Inc., Suite 330, 1201 5th Street S.W, Calgary, Alberta, Canada, T2R-0Y6, 1998.

[9] R. Polikar, "Esemble based systems in decision making," *IEEE Circuits and Systems Mag.*, vol. 6, no. 3, pp. 21–45, 2006.

[10] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies using Traffic Feature Distributions," in *ACM SIGCOMM, Philadelphia, PA, August 2005*, New York, NY, USA, 2005, pp. 217–228, ACM Press.

[11] J. Newsome, B. Karp, and D. Song, "Paragraph: Thwarting signature learning by training maliciously," in *Recent Advances in Intrusion Detection*. 2006, pp. 81–105, Springer.

[12] L. Ertoz, E. Eilertson, A. Lazarevic, P. N. Tan, V. Kumar, J. Srivastava, and P. Dokas, "MINDS - Minnesota Intrusion Detection System," in *Next Generation Data Mining*. 2004, MIT Press.

[13] K. Xu, Z.L. Zhang, and S. Bhattacharrya, "Reducing Unwanted Traffic in a Backbone Network," in *USENIX Workshop on Steps to Reduce Unwanted Traffic in the Internet (SRUTI)*, Boston, MA, July 2005.

[14] Avinash Sridharan, T. Ye, and Supratik Bhattacharyya, "Connectionless port scan detection on the backbone," in *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, april 2006, pp. 10 pp. –576.

[15] Martin Rehak, Michal Pechoucek, Karel Bartos, Martin Grill, and Pavel Celeda, "Network intrusion detection by means of community of trusting agents," in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007 Main Conference Proceedings) (IAT'07)*, Los Alamitos, CA, USA, 2007, IEEE Computer Society.