ARAKIS – AN EARLY WARNING AND ATTACK IDENTIFICATION SYSTEM

Piotr Kijewski[1]
CERT Polska/NASK

ABSTRACT

The paper outlines the concept of an early warning and attack identification system, ARAKIS, being developed by CERT Polska. The system is meant to detect and identify the characteristics of large scale novel threats, such as self-propagating malicious code and other automated attacks that span across multiple sites. Its goals also include the automated creation of attack signatures for dissemination to intrusion detection/prevention systems and providing attack statistics. The paper presents the rationale behind the system, data sources used, architecture and current stage of development.

1. INTRODUCTION: TRENDS IN LARGE SCALE MALICIOUS CODE ATTACKS

The last few years have seen an increase in self-propagating malicious code and automated attacks against organizations connected to the Internet. These threats are mostly indiscriminate, directed against the Internet as whole. An example is the SQL Slammer worm, which, by exploiting a flaw in Microsoft SQL Server, succeeded in infecting over 75 000 hosts by searching through 90% of the Internet address space in under 10 minutes [1]. However, recent threats, for example, the Bugbear.B virus, had functionality that allowed it to search for a specific type of victim. After infection, Bugbear.B checked if the victim's address corresponded with addresses of over 1000 financial institutions in the world. In case of a match it searched the system for passwords and other sensitive information [2].

It is therefore possible that future threats will distinguish between targets and focus on certain groups of networks, on the basis of different criteria, such as, for example national, geographical or "functional" relevance. These targeted attacks will be more difficult to detect and counter, as by definition, they will not be observed by the entire Internet community.

Another noticeable trend is the ever shortening window between the time of a vulnerability announcement and exploit release, as well as a subsequent worm (or some other automated tool) appearance. For example, the Witty worm appeared only a day after the vulnerability it exploited was publicized [3] giving network administrators very little time to patch their systems.

[1] E-mail: piotr.kijewski@cert.pl

This has exacerbated the weakness of current, mostly signature based, intrusion detection and prevention technology, as well as antivirus systems. These systems base their knowledge on rules of known attacks. In a sense, they are historic in nature. Their effectiveness is based on the speed with which a new rule is designed, its correctness and the swiftness of its deployment. Such systems, working alone, are unlikely to successfully perform their function in a situation where self-propagating code makes use of a zero-day exploit.

The distinction between worms, viruses, trojans, backdoors, botnets is becoming increasingly blurred. A threat is often a blend of the above. These new threats propagate in many different ways. They can only be adequately understood through correlation from many different types of data sources. As their code grows more complex, it also becomes more susceptible to bugs. Initial code analysis sometimes fails, as due to the bugs the threats function not quite as their author intended.

2. PROJECT GOALS

ARAKIS[2] is a project initiated by CERT Polska to address the above mentioned trends. The project's goal is to create a system that will aid in the **automated detection**, **analysis** and **response** to large scale malicious code attacks and other automated attacks against Polish networks, possibly those that can be considered a part of the Polish national critical infrastructure. In particular, the system is expected to:

- enable the automated detection of new threats, in particular those that make use of novel network attacks,
- automate the process of attack analysis, allowing for the identification and description of novel attacks,
- develop a method of automated signature creation and dissemination to firewalls and intrusion detection and prevention systems,
- develop an automated process for the comparison of trends across multiple administrative domains,
- improve network situational awareness,
- serve as an aid in general incident handling,
- provide network attack statistics.

The project is expected to provide a unique view of attacks against Polish networks and supply data that can be used as a basis for comparison with attack data from other countries and systems. Our intention is to concentrate on developing **practical** methods of achieving the above goals.

[2] The acronym ARAKIS stands for "**A**g**R**egacja, **A**naliza i **K**lasyfikacja **I**ncydentów **S**ieciowych" in Polish, which can be translated to „Aggregation, Analysis and Classification of Network (Security) Incidents". Frank Herbert's DUNE, the planet Arrakis and its famous worms, also had an impact on the project title.

3. UNDERSTANDING THREATS, TARGET SELECTION AND PROPAGATION METHODS

For effective techniques to counter threats to be developed, the nature of malicious code attacks has to be understood. There have been many studies into the nature of malicious code conducted in recent years. One of the most recent concentrated on developing a taxonomy of worms [4]. The two classification criteria of those presented in the taxonomy that we feel are most relevant to our project are target selection and method of propagation. This is because these factors have a significant impact on the choice of data sources that can be used by a detection, analysis and response system. They are also not just relevant to worms but malicious code attacks and automated attacks in general.

The taxonomy identifies a number of target selection methods used by worms – scanning, pre-generated target lists, externally generated target lists, internal target lists and passive. A worm may use more than one combination of these. Scanning, which can be, for example sequential, or random involves probing a set of addresses to identify vulnerable hosts. It is the most popular form of target selection, used by many famous worms, including Code Red I and II, Nimda, Slammer, Blaster and Witty. It is also one of the easier types to detect, because scanning shows up as highly anomalous traffic. Pre-generated target lists can be compiled by attackers to speed up initial propagation or to specifically target a group of institutions. Externally generated target lists can be created. For example, a worm may use a search engine, such as Google to identify vulnerable web servers. A worm may also use internal target lists – information about a local network topology gathered from an infected host. Finally, a worm may be passive. In this case it waits for the victim to contact it in order to infect him and spread.

Malicious code can be self-carried. In this case, the malicious code is transmitted as part of the infection process. Sometimes a second channel is needed to be opened to transfer a threat, for example, the Blaster worm used the RPC DCOM exploit to gain access to a host, but opened a TFTP channel to the source of infection in order to transfer the body of the worm. Malicious code can also be embedded as part of a normal communication channel, making it much more difficult to identify it as anomalous.

4. ARAKIS DATA SOURCES OVERVIEW

The different selection and propagation methods make correlation the basis for detecting novel large scale attacks. The fact that many sensors across different sites see a similar trend makes detecting anomalies easier, and lowers the amount of false positives. The complexity of threats, different target discovery and propagation techniques demands different data sources be used. What follows is a discussion of the data sources used in the project. While one could imagine that the perfect system would utilize detectors residing on nearly every network device, gathering information directly from a network, system or application level, such a system is beyond the scope of the project. Instead, we concentrate on data sources that can be found or relatively easily deployed on most networks.

Firewalls

Firewalls[3] and the packets they drop are one of the simplest data sources for intrusion detection. They are useful for a number of reasons. Firstly, firewalls are used to protect production networks, so information from them is information about real threats against a production network. Secondly, the fact that a firewall rejects some packets is important, because it means a network administrator deemed such attempts as unwanted. A firewall thus makes a significant contribution to the detection process, because it easily identifies anomalous packets. For example, a rise in dropped packets to destination port 21/TCP may by caused by a new worm attempting to propagate through an application residing on that port. Finally, firewalls are used by virtually every larger institution connected to the Internet, thus allowing for a larger set of destination addresses to be observed.

Firewalls are especially useful in detecting threats that utilize scanning as their target discovery mechanism. They are less effective in detecting attacks that employ other forms of target discovery. For example, a worm that spreads through a P2P network, passively waiting for a human to transfer it will not be detected by this source. In theory, the same could be said of mass e-mail viruses or worms. These propagate by gathering e-mail addresses from a compromised host (a form of internal target list discovery). By doing so they should avoid firewall filters. However, that is not always the case – outdated e-mail addresses sometimes point to firewalled hosts. Thus, monitoring dropped packets from firewalls sometimes allows for the detection of e-mail threats as well.

Honeypots

Honeypots are resources placed on a monitored network with the assumption that they may be compromised by an attacker, allowing the attacker's actions to be studied. In ARAKIS, honeynets[4] serve as the primary source of information in the process of attack analysis. Their advantage as an information source lies in the fact that they observe mostly malicious traffic. This means that the detection process is very simplified, and effort can be concentrated on analysis. Honeypots are useful in capturing malicious code that uses scanning as target selection method, although they can potentially capture malicious code using other active target selection methods too. Broadly speaking, honeypots can be divided into two categories: low-interactive, simply emulating a service and high-interactive, in the form of a real

[3] This includes routers with ACLs

[4] Networks of honeypots

operating system that can be compromised. As we initially intend to focus on the observation on attacks at the network level and the initial attack vector, ARAKIS assumes the deployment of low-interactive honeypots, such as *honeyd* [5].

Honeynets may also attract non-automated attacks, which may allow for the identification of novel exploits before they are utilized in a large scale attack. A potential weakness is that they are not production networks, and that they may be identified and avoided by an attacker. This can be partly mitigated if the honeynet uses addresses assigned to a production network as well.

Netflow

Netflow is another data source that will be used by the system. Netflow data from routers provides a unique view as to what is happening on an ISP's WAN. Netflow is the primary source of information concerning Distributed Denial of Service (DDoS) attacks. Its second contribution is the fact that it can be used to observe a complete attack process, such as the opening of a secondary channel for data transfer (employed for example by the Blaster worm), because it observes real networks, not honeynets. It allows for a better understanding of propagation carriers and distribution mechanisms used by a threat. The main drawbacks associated with netflow data is its huge amount, and the fact that it provides little information about packet headers and none about the payload content.

Knowledge based systems

ARAKIS is meant to be a system that can give meaning to the collected security information and function as a knowledge base, specifically tailored for large scale malicious code attacks. To this end, existing knowledge based systems, such as network intrusion detection systems and antivirus systems can be used in a supportive role. Despite their drawbacks mentioned earlier, these systems are a useful source of information about known attacks.

5. SYSTEM ARCHITECTURE OVERVIEW

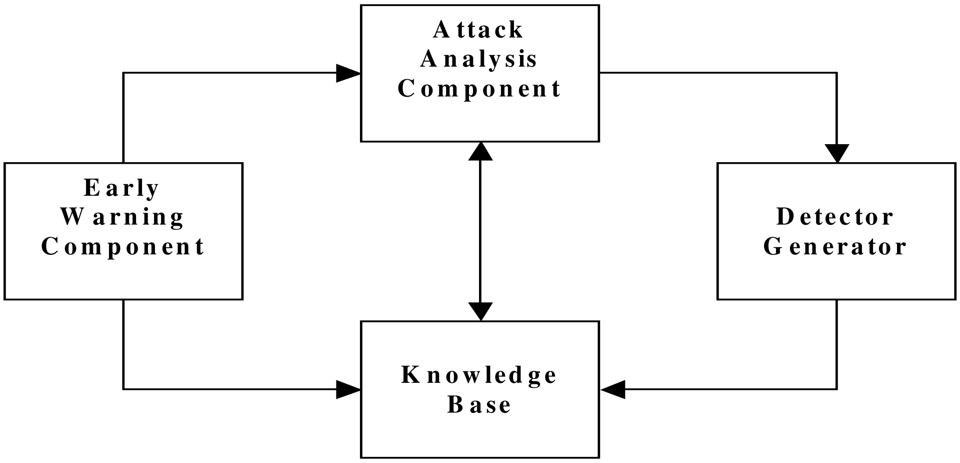

Figure 1. Information flow between different ARAKIS components

The prototype system is being built around distributed sensors that supply information to a central repository for analysis. A more sophisticated architecture is considered in the future, but as of now, the focus is on researching and developing effective analysis techniques. The functionality architecture consists of four components: an early warning component, an attack analysis component, a detector generator and a knowledge base component. The information flow between the modules is shown in Figure 1.

Early Warning Component

The early warning component is responsible for:

- detection of a new threat,
- issuing a timely alert,
- providing support data to the attack analysis component.

The early warning component's role is to analyze network traffic looking for anomalies that may be indicative of an emerging threat. If an anomaly is detected some preliminary analysis is done in order to identify some basic characteristics of the new threat. This information is then relayed to the attack analysis component. The goal of the component is fast detection of changes in network activity, not analysis, even though potentially both the detection and analysis components may use the same data sources.

The initial implementation of the early warning component makes use of sensors of firewall dropped packets as its data source. Simple trend analysis (for example, an increase in distinct source addresses observed on a port) is done based on the data, which is usually limited to source IP, destination IP, source port, destination port, protocol and date information[5]. If a new trend is observed, information about the affected destination ports, observed sources, their autonomous system, and country of origin is relayed to the attack analysis component. The alerts and supporting data are also published on a web page, in the form of graphs, tables and maps.

Attack Analysis Component

The attack analysis component is responsible for creating an understanding of a new threat:

- collecting attack data and organizing it into preliminary structures facilitating analysis,
- identifying if the threat and attack in question is novel,
- describing the full attack scenario.

Analysis of a threat can be performed at various levels - network, host, and code disassembly. The initial focus of the project is on automating the analysis on the network level. Analysis will be done through a series of tests, ranging from simple tests that can be completed on-line to more complex and time consuming off-line tests. For example, the simplest test may involve checking for new combinations of scanned destination ports or novel packet content, more complex may involve guesses as to the character of exploits used (analyzing packet payload content, not just headers) or attempt to describe the full attack scenario, packet by packet. Analysis will be supported by data from the knowledge base, which contains descriptions of previously seen threats and attacks. We intend to investigate data mining and machine learning techniques that could be applied to this process. Currently we have developed a sniffer type program that monitors packets on the wire, and organizes them into preliminary structures in memory to facilitate analysis.

For attack analysis, honeynets are intended as the primary data source. Information from the early warning component concerning the ports attacked and attack sources is intended to make the honeynets focus on trends as seen by firewalls, and speed up the initial analysis process. The information supplied by firewalls may also be used to open up new ports on honeynets if necessary. Intrusion detection systems such as *snort* [6], operating in the honeynet environment contribute to the recognition of known attacks.

Netflow plays a supporting role in the analysis process. We plan to apply patterns discovered from the honeynet data source to netflow data in an attempted to extract more knowledge about the behavior of infected hosts.

[5] Some firewalls provide more information, such as packet size, TCP flags, sequence numbers etc.

Detector Generation Component

Automated response is the end goal of the system. The analysis component is expected to identify threat and attack characteristics, and possibly the full attack scenario. This is necessary for a human analyst to understand the nature of a threat, but insufficient to institute an automated response to block a threat. What is needed is a precise rule that can be used by firewalls, intrusion detection/prevention systems. Based on data supplied by the analysis component, the detector generator's goal is to discover such rules.

Knowledge Base

The knowledge database will contain labeled data about identified threats, attacks, and their characteristics, as well as signatures proposed by the system. It is used to support the attack analysis module, and identify if the threats and attacks seen are novel. The data is expected to be under constant review of a human expert. One of the challenges here will be to develop formats for the descriptions of automated threats.

6. RELATED WORK

There has been a lot of interest lately in creating practical intrusion detection systems that can operate in large scale networks. The following are projects which we feel are of particular relevance to our work. DSHIELD [7] is a well-known system that collects dropped packet data from firewalls and port scanning logs from intrusion detection systems, located all over the world. It is currently limited to observing trends in port activity, which may be indicative of new wide-spread threats, and does not provide automated descriptions of attacks or automatically suggest signatures. Under the Honeynet project [8] a set of distributed honeynets is planned to be connected to a central repository, to enable attack correlation and analysis. Currently however, most of the effort appears focused on the development of sophisticated, easy to deploy honeypots and monitoring techniques. AirCERT [9] is a CERT/CC project, which aims to create a scalable distributed system for sharing security event data among administrative domains. A set of tools has already been published that enable normalized data exchange. The eCSIRT.net project [10] uses the Prelude IDS [11] as the basis for establishing a distributed sensor network of honeypots. Attacks are only observed through IDS signatures, which limit the ability to detect and describe novel attacks. Rather than reinvent the wheel, we intend do adapt some of the software and solutions used in these projects where they apply, and focus on analysis instead.

7. CONCLUSION

Detecting novel network attacks at an acceptable false alarm rate has been the "holy grail" of intrusion detection. For certain threats described in this paper, practical automated detection and identification appears feasible. ARAKIS is intended as a framework for detecting these groups of threats. By developing ARAKIS, insight may be gained as to what is needed to detect and identify other and future threats. To accomplish this, other data sources will need to be added. Even though the focus is on external threats to a network, a system that learns about such threats is capable of contributing to the security of an internal network, through the development of threat signatures that can be used by appliances protecting segments of a local network.

REFERENCES

[1] David Moore, Vern Paxson, Colleen Shannon, Stuart Staniford, Nicholas Weaver. The Spread of the Sapphire/Slammer Worm, 2003, http://www.caida.org/outreach/papers/2003/sapphire/

[2] Symantec Security Response. W32.Bugbear.B@mm, http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear.b@mm.html

[3] Colleen Shannon, David Moore. The Spread of the Witty Worm, 2004 http://www.caida.org/analysis/security/witty/

[4] Nicholas Weaver, Vern Paxson, Stuart Staniford, Robert Cunningham. A Taxonomy of Computer Worms, October 2003, http://www.cs.berkeley.edu/~nweaver/papers/taxonomy.pdf

[5] http://www.honeyd.org/

[6] http://www.snort.org/

[7] DSHIELD - Distributed Intrusion Detection System, http://www.dshield.org/

[8] The Honeynet Project, http://www.honeynet.org/

[9] AirCERT, http://aircert.sourceforge.net/

[10] eCSIRT.net – The European CSIRT Network, http://www.ecsirt.net/

[11] Prelude Hybrid IDS project, http://www.prelude-ids.org/