DarkNOC: Dashboard for Honeypot Management

Bertrand Sobesto, Michel Cukier
Clark School of Engineering
University of Maryland
College Park, MD, USA
{*bsobesto, mcukier*}*@umd.edu*

Matti Hiltunen, Dave Kormann, Gregg Vesonder
AT&T Labs Research
180 Park Ave.
Florham Park, NJ, USA
{*hiltunen, davek, gtv*}*@research.att.com*

Robin Berthier
Coordinated Science Laboratory
Information Trust Institute
University of Illinois
Urbana-Champaign, IL, USA
rgb@illinois.edu

Abstract

Protecting computer and information systems from security attacks is becoming an increasingly important task for system administrators. Honeypots are a technology often used to detect attacks and collect information about techniques and targets (e.g., services, ports, operating systems) of attacks. However, managing a large and complex network of honeypots becomes a challenge given the amount of data collected as well as the risk that the honeypots may become infected and start attacking other machines. In this paper, we present DarkNOC, a management and monitoring tool for complex honeynets consisting of different types of honeypots as well as other data collection devices. DarkNOC has been actively used to manage a honeynet consisting of multiple subnets and hundreds of IP addresses. This paper describes the architecture and a number of case studies demonstrating the use of DarkNOC.

1 Introduction

Because of the value of the data they store and the resources they provide, information systems become targets for attackers and must be protected. To better secure computer systems from external threats, security researchers aim to understand attackers and the different techniques they use to compromise computers and achieve their goals. One possible approach is to use a target computer, called a honeypot, which is not used by normal users. Therefore, all the activity towards this computer can be considered malicious.

Individual honeypots or networks of honeypots have been used to conduct various studies of attackers [1, 9] and analysis of cyber crimes such as unsollicited electronic mails, phishing [10], identity theft and denial of service. The computer security community has used honeypots to analyze different techniques deployed by the attackers to reach their objectives. Attackers' arsenal includes distributed denial of service [24], botnets [2], worms [11] or SPAM [15]. However few studies focus on the usage of honeypots data to help network administrators to better protect their production networks. Honeypot deployment is challenging and the architecture of such networks is complex. For example, distributed honeynets require secure tunnels and different levels of protection must be in place to ensure a total containment of attacks targeting the honeypots. In addition, honeynets require constant monitoring to guarantee that protection systems (for example firewalls, traffic shappers) and data collection are operating correctly. Depending on the size of the honeynet, the volume of data collected can be important and impacts significantly data processing and extraction. To be integrated as a security tool, honeypots data must be presented and translated in meaningful way to network administrators.

In this paper, we introduce DarkNOC, a solution designed to efficiently process large amount of malicious traffic received by a large honeynet, and to provide a user-friendly Web interface to highlight potential compromised hosts to security administrators, as well as to provide the overall network security status. DarkNOC is used to manage the UMD honeynet, a network of 2,000 honeypots from which information about attacks is continuously extracted and provided to the security team to help them better protect the production network.

The rest of the paper is organized as follows. In Section 2, we provide an overview of the architecture and operation of DarkNOC. In Section 3, we describe the outputs and views provided by the DarkNOC. We provide a number of case studies using DarkNOC in Section 4. Finally we review the related work in Section 5, we provide some remarks on future work in Section 6 and conclude the paper in Section 7.

2 DarkNOC Architecture

This section describes what DarkNOC does, how it collects data, and its internal structure.

2.1 System Architecture

DarkNOC manages multiple types of honeypots and information sources as illustrated in Figure 1. The UMD honeynet consists of low interaction honeypots (LIHs) such as Nepenthes [3] as well as high-interaction honeypots (HIHs) consisting of virtual or physical machines running real operating systems, applications, and services [5]. The UMD honeynet supports multiple subnets consisting of IP addresses contributed by different organizations participating in the research. DarkNOC collects multiple sources of information from different devices (e.g., NetFlow from Gateway, Snort events from Snort Sensors [20], and malware from Nepenthes), analyzes the data, and presents it to users in an efficient and actionable manner. The details of the data views and their use in analyzing security incidents are discussed in Sections 3 and 4.

The current information sources consist of the following:

- NetFlow Data: DarkNOC uses nfdump[1] to extract NetFlow data collected on the main gateway of the honeypots. The flow data provides enough information to determine the number of attackers, the different source and destination IP addresses, and the different source and destination ports. Specifically, each NetFlow record summarizes communication between two network end points (defined by the IP addresses and port numbers of the end points) including the time, duration, and numbers of bytes and packets (see example below), but does not contain any payload information (i.e., content of the messages transmitted).

```
Date flow start      Duration  Port  Src IP:Port  -> Dst IP:Port  Packets Bytes Flows
2010-02-09 06:43:... 4294966.937 TCP  218.8.251.187:20347 -> x.x.x.x:80  2  94   1
2010-02-09 06:43:... 4294966.977 TCP  218.8.251.187:20347 -> x.x.x.x:80  2  94   1
```

[1] http://nfdump.sourceforge.net/

- Snort Events: Snort [20] is an Intrusion Detection System (IDS) for detecting attacks and potential intrusions. Snort provides information about the types of attacks used against the honeypots.

- Malware Collection: Nepenthes acts as a passive malware collector by emulating common service vulnerabilities and allowing attackers to inject the malware binaries. Nepenthes provides a log of each malware submission containing information such as the date and the vulnerability used but also the binary injected. This allows DarkNOC to see what kinds of malware are successfully uploaded, the security signatures, and port used. It also allows to measure the efficiency of the security solution protecting the network.

2.2 DarkNOC Software Architecture

The design of the DarkNOC software architecture was driven by the following constraints:

- The aesthetics from the user's point of view: The user interface should be easy to access and the important data should be automatically highlighted. This interface should be highly portable so that users can use different operating systems and access the system from different geographic locations (i.e., not tied to one dedicated machine).

- Speed: The user interface must be fast and the user should not have to wait for the results to be displayed. Processing high volumes of data can be time consuming and if the processing is started only when the user requests a data view, the response time may not be satisfactory. Therefore, our system uses data pre-processing when possible to ensure fast response.

- Data validity: The data displayed should be reasonably up to date and reflect the current activity.

To meet these requirements, the application software has been divided into three different parts: 1) a graphical Web front-end, 2) back-end, and 3) alerting module. The front-end generates a Web page displaying the different information. The back-end extracts the necessary data from the flows and creates the different graphs.

Back-end Module: Written in Perl, the back-end module is a background process that updates the information displayed by the front-end every 5 minutes based on the NetFlow data. The separation of flow processing from the display was necessary to guarantee a fast response time at the user interface, because the extraction of flow

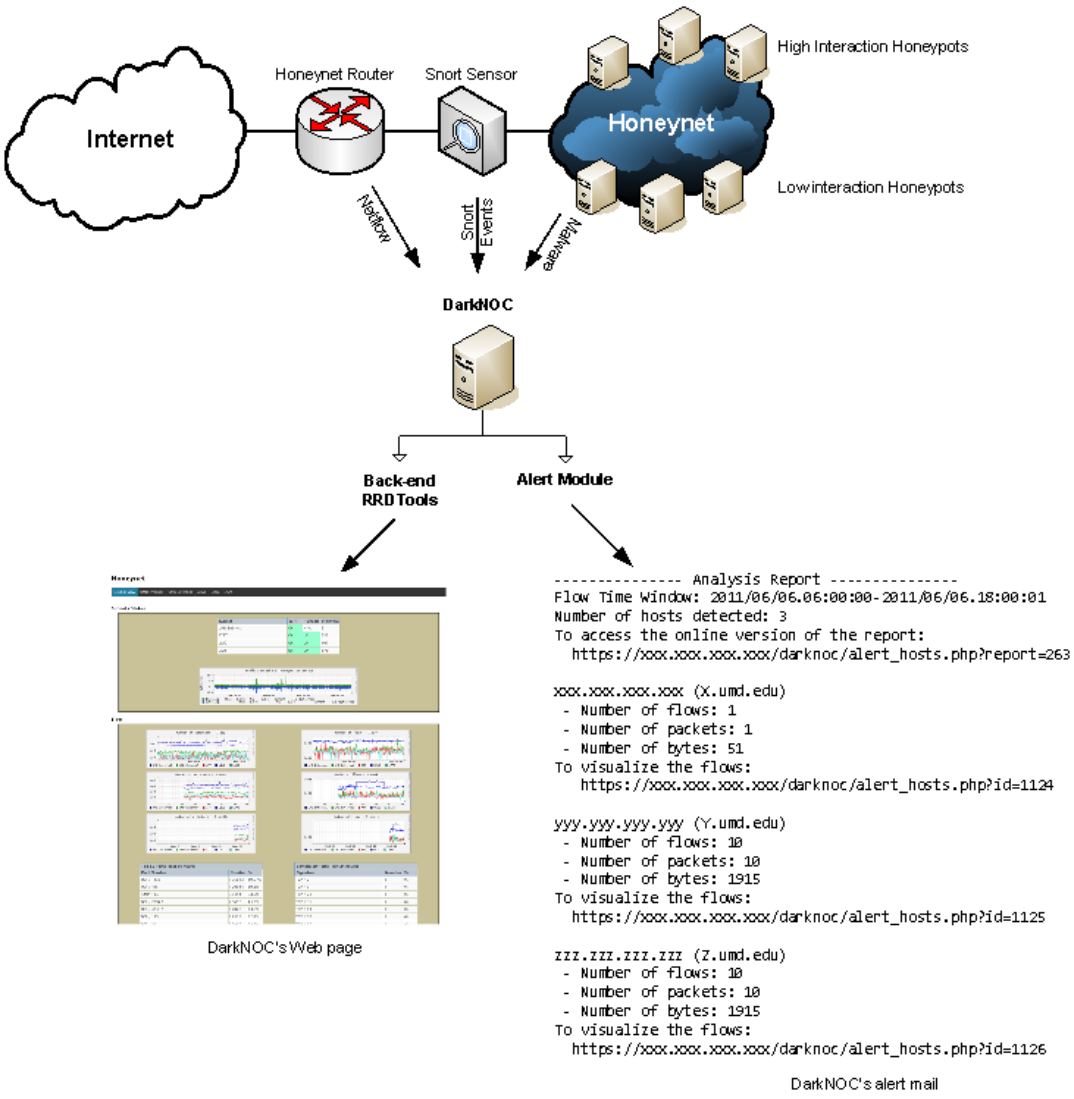

Figure 1: System architecture

data can be time consuming. Since the flow data is updated every 5 minutes by the flow collector, a continuous live update of the displayed views is unnecessary. However it requires the tool to process the new flow files within 5 minutes. DarkNOC provides information for the last 24 hours and the last 5 minutes. Two different processes generate the 24 hours and 5 minutes statistics. For about 2,000 IP addresses, an average of 15,995 flows are generated every 5 minutes representing about 5 million flows per day. It takes an average of 7.4 seconds to process a newly created flow file. Given this number, DarkNOC is able to process almost a hundred times more flows within 5 minutes. Generating the statistics on the last 24 hours is computationally more expensive and longer. It takes an average of 130 seconds. However, it is not necessary for this process to finish within 5 minutes. A lock file prevents multiple executions of this process at the same time. For each subnet and the global view, the back-end generates the different graphs, the list of destination ports, the list of attackers and the list of targeted honeypots. The graphs are created using RRDTool[2], an open source tool for storage and retrieval of time series.

Graphical User Interface: The graphical user interface organizes the different data necessary to present a summary of the honeypots activity. Web technologies such as the PHP language and Cascading Style Sheets are used. A Web page is extremely portable and requires no configuration on the client side. Figure 2 shows the homepage of DarkNOC. The content is described in Section 3. The graphical user interface first provides a global

[2]http://oss.oetiker.ch/rrdtool/

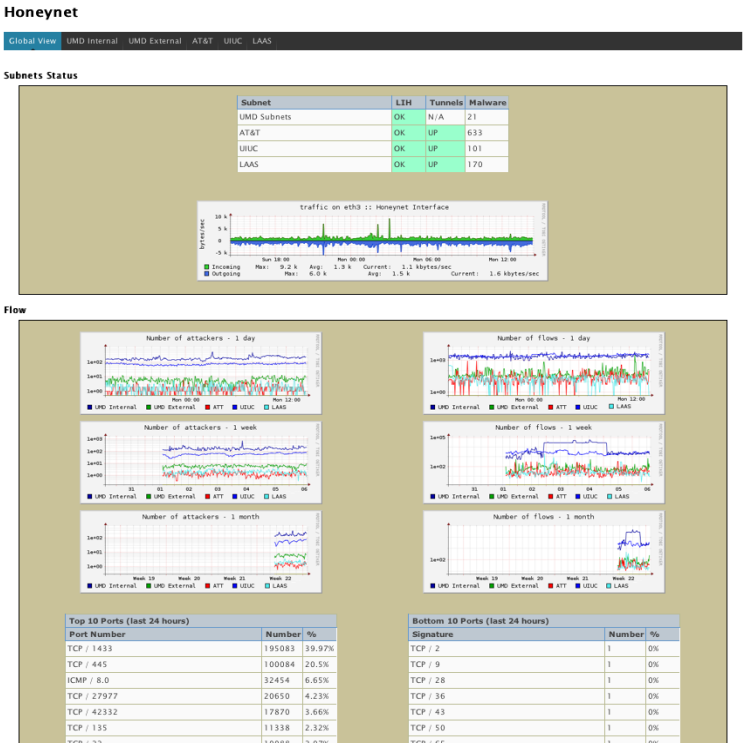

Figure 2: DarkNOC's graphic user interface

view of the activity of the honeypots: the data displayed includes all the subnets. The user has then the possibility to reduce the scope of analysis to one subnet. To prevent unauthorized access, the application uses an HTTP authentication over SSL to protect DarkNOC's directory on the Web Server. Apache is configured to authenticate users against an LDAP server where all accounts are centralized. User objects belonging to the group *DarkNOC* have access to the application. Because of legal and confidentiality reasons it is necessary to filter the information displayed by DarkNOC. Once authenticated DarkNOC retrieves the user name stored in the `$_SERVER['PHP_AUTH_USER']` variable and matches it with the user's table in the database to determine which subnets to display or not. If the user is allowed to access more than one subnet, DarkNOC will reflect the user's rights in the global view but also in the subnet selector. If the user has access to a single subnet, the subnet will be automatically selected with no possibility to select another one.

Alerting Module: The alerting module is a process executing a specific query on the flow data. The results are sent by email to a specific group of users. Users have the possibility to create their own flow query based on the nfdump filter syntax and to specify the recipients of the alerts. The module is currently launched twice a day: at 6:00 AM and at 6:00 PM. It can be executed more frequently if more real-time alerts are required.

3 Display Description

The layout of the graphical user interface of DarkNOC presented in Figure 2 organizes the different pieces of information gathered from the most global and important to the most detailed concerning the current activity of the honeypots. The user interface of DarkNOC has been developed to ease the comparison of the different sources of information and the comparison of the different subnets.

The Web page provided by DarkNOC is divided into three different sections: 1) status of the subnets, 2) flow-based information, and 3) Snort events. Each section will provide information that will reduce the number of possible explanations when an anomaly in the traffic is identified in DarkNOC. The first screen provided is a global view of the honeypots activity. The user can select a specific subnet to drill-down to a more detailed view of the subnet activity.

3.1 Subnet Status and Network Traffic

The first part of the Web page shown in Figure 3 is composed of a table giving the status of the low interac-

Subnets Status

Subnet	LIH	Tunnels	Malware
UMD Subnets	OK	N/A	21
AT&T	OK	UP	633
UIUC	OK	UP	101
LAAS	OK	UP	170

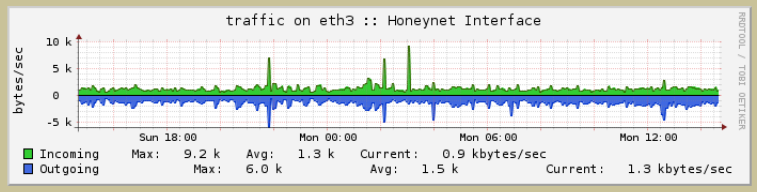

Figure 3: Subnets status section

Top 10 Ports (last 24 hours)		
Port Number	**Number**	**%**
TCP / 1433	198939	40.03%
TCP / 445	101807	20.49%
ICMP / 8.0	32848	6.61%
TCP / 27977	21520	4.33%
TCP / 42332	18191	3.66%
TCP / 135	11563	2.33%
TCP / 22	10105	2.03%
TCP / 80	5204	1.05%
UDP / 5060	5119	1.03%
TCP / 139	4914	0.99%

Bottom 10 Ports (last 24 hours)		
Signature	**Number**	**%**
TCP / 2	1	0%
TCP / 9	1	0%
TCP / 28	1	0%
TCP / 36	1	0%
TCP / 43	1	0%
TCP / 50	1	0%
TCP / 65	1	0%
TCP / 66	1	0%
TCP / 67	1	0%
TCP / 71	1	0%

Figure 4: Top and bottom 10 transport ports targeted

tion honeypots (LIH) running Nepenthes, the status of the tunnels to different organizations, and the number of malware collected for each subnet since the initialization of DarkNOC. The notion of a tunnel is specific to the UMD honeynet. It allows to redirect the network traffic from remote locations to the honeypot network transparently. Hence, it is possible to use other participating organizations' IP addresses. A graph representing the incoming and outgoing traffic in bytes per seconds is included in the status section as well. This section provides essential indications on the state of the main components of the UMD honeynet, i.e. tunnels and main gateway. The graph gives an overview of the UMD honeynet infrastructure load and can help to detect anomalies in the traffic.

3.2 NetFlow Data

The NetFlow section provides information extracted from the NetFlow data collected at the edge of the honeypots network. Figure 5 presents a graph showing the number of attackers over time for each subnet of the honeypot network. Each unique IP address that does not belong to the honeypots is considered a unique attacker. The graphical user interface provides several graphs that display the number of attackers at different time scales: one day, one week, and one month. Figure 6 presents a graph showing the number of flows over time for each subnet of the honeypot network. Separate graphs are used to display the number of flows at different time scales.

These two graphs shown in Figures 5 and 6 make it easy to observe the activity of the honeypots for each subnet. Comparing the numbers of flows and attackers can reveal attack characteristics. For example, an increase of the number of flows while the number of attackers remains relatively steady means that one or several offenders may have launched an attack that generates large amounts of flows such as port scanning and brute-force activities. It can also mean that a large network behind a network address translation system is compromised and targeting the UMD honeynet. DarkNOC also makes it easy to compare trends between the different

subnets. For example, it is straightforward to identify peaks in the number of attackers or flows that occur at the same time in different subnets, as well as changes in the attacks directed to only one of the subnets, indicating a targeted attack.

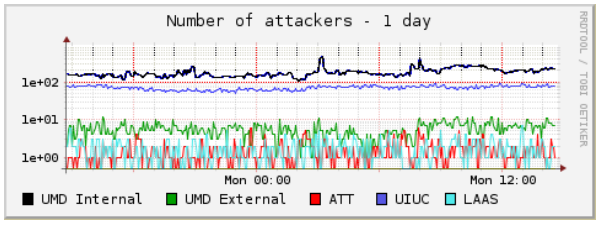

Figure 5: Number of attackers

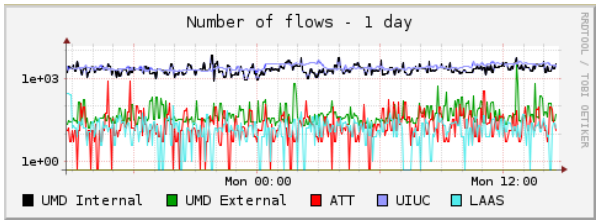

Figure 6: Number of Flows

The tables in Figure 4 show the top and bottom 10 ports targeted by the attackers during the last 24 hours. For each port, the number of flows and the percentage of the total number of flows are provided. It makes it easy to identify the most *popular* services and to protect the network accordingly. The severity of an attack is not related to the number of flows it will generate. Attacks towards common ports tend to hide smaller attacks against less popular ports. This is why we also decided to display the bottom 10 ports targeted.

Finally, Figure 7 represents a word cloud of the top 20 attackers' IP addresses. The top 20 IP addresses are determined using the number of flows involved in the communications between the attacker and the honeypots. The size of the font displaying the IP address reflects the number of flows generated for that IP address. The same representation is used for the top 20 targeted honeypots. These word clouds are updated every 5 minutes using a 24-hour window. The IP addresses presented in the word clouds are clickable: The user can obtain the lists of honeypots contacted, services and Snort events related to the selected IP address in a separate window. Since the honeypot network often hosts different experiments with different configurations, the port tables and the targeted honeypots make it possible to determine what is attracting the attackers the most.

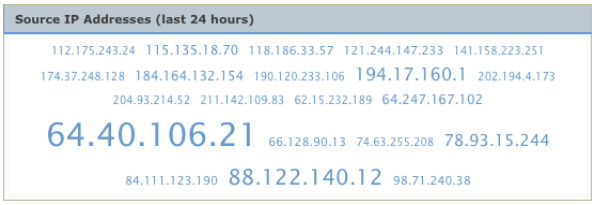

Figure 7: Attacker word cloud

3.3 Snort Data

Last 10 Snort Events

Timestamp	Signature	Source	Destination
2011-06-06 14:41:53	stream5: Limit on number of overlapping TCP packets reached	xxx.xxx.xxx.xxx	174.77.190.64
2011-06-06 14:41:53	stream5: Bad segment, overlap adjusted size less than/equal 0	xxx.xxx.xxx.xxx	174.77.190.64
2011-06-06 14:41:50	ICMP PING NMAP	94.248.15.15	xxx.xxx.xxx.xxx
2011-06-06 14:41:45	stream5: Limit on number of overlapping TCP packets reached	xxx.xxx.xxx.xxx	190.11.17.22
2011-06-06 14:41:45	stream5: Bad segment, overlap adjusted size less than/equal 0	xxx.xxx.xxx.xxx	190.11.17.22
2011-06-06 14:41:45	stream5: Limit on number of consecutive small segments reached	xxx.xxx.xxx.xxx	190.11.17.22
2011-06-06 14:41:44	stream5: Limit on number of overlapping TCP packets reached	xxx.xxx.xxx.xxx	78.187.81.52
2011-06-06 14:41:44	stream5: Bad segment, overlap adjusted size less than/equal 0	xxx.xxx.xxx.xxx	78.187.81.52
2011-06-06 14:41:44	stream5: Limit on number of consecutive small segments reached	xxx.xxx.xxx.xxx	78.187.81.52
2011-06-06 14:41:43	stream5: Limit on number of overlapping TCP packets reached	xxx.xxx.xxx.xxx	94.97.113.112

Figure 8: Last 10 Snort events table

The Snort section presents information about the Snort alerts.

Figure 8 shows a table of the last 10 Snort events collected on the honeypot network. This table allows honeypot administrators to immediately identify attacks generating high volumes of traffic. For example, a brute-force attack against a Microsoft SQL server will generate a spike in the traffic curves and the corresponding events will appear immediately in this table.

The graph in Figure 9 provides a trend in the number of Snort events recorded the current day, the past few days, and the past few weeks.

Figure 10 shows the top and bottom 10 Snort signatures tables. The tables provide the signature name, the number of events for each signature and the percentage. Large scale attacks such as port scanning or brute-force attacks may generate several events. As a consequence, smaller but still important attacks may not appear in the top 10 signatures. This is why the bottom 10 Snort signatures are also provided. As an example, consider the snort signature *SHELLCODE NOOP* shown in the Bottom 10 Snort events of Figure 10. This signature indi-

Top 10 Snort Events (last 24 hours)		
Signature	**Number**	**%**
snort: "SQL sa brute force failed login unicode attempt"	56932	54.55%
"MS-SQL SA brute force login attempt TDS v7/8"	23851	22.85%
stream5: Bad segment, overlap adjusted size less than/equal 0	8357	8.01%
stream5: Limit on number of overlapping TCP packets reached	7805	7.48%
stream5: Limit on number of consecutive small segments reached	2325	2.23%
MISC MS Terminal server request	1519	1.46%
ICMP PING NMAP	1271	1.22%
ssh: Protocol mismatch	801	0.77%
stream5: Data sent on stream not accepting data	308	0.3%
stream5: Packet missing timestamp	305	0.29%
More...		

Bottom 10 Snort Events (last 24 hours)		
Signature	**Number**	**%**
snort: "SPECIFIC-THREATS ASN.1 constructed bit string"	1	0%
WEB-IIS WEBDAV nessus safe scan attempt	7	0.01%
ICMP Source Quench	7	0.01%
ICMP L3retriever Ping	15	0.01%
snort: "SHELLCODE base64 x86 NOOP"	19	0.02%
ICMP Destination Unreachable (Communication with Destination Host is Administratively Prohibited)	25	0.02%
ftp_pp: Invalid FTP command	28	0.03%
"POLICY RDP attempted Administrator connection request"	29	0.03%
stream5: TCP Timestamp is outside of PAWS window	30	0.03%
ICMP Destination Unreachable (Communication Administratively Prohibited)	34	0.03%
More...		

Figure 10: Top and bottom 10 Snort signatures

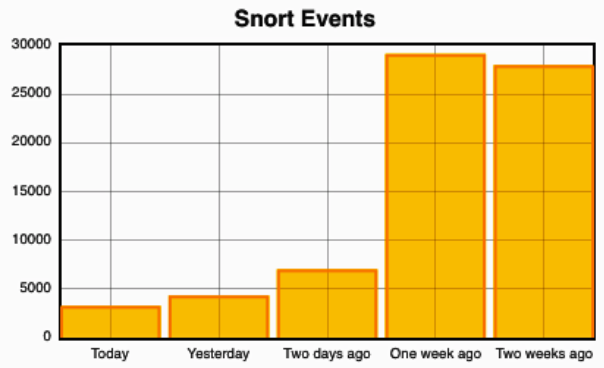

Figure 9: Snort events graph

cates attempts to upload a malicious shellcode.

In the following example, the Snort IDS alerts show a possible injection of malicious code on an emulated Web server:

```
04/15-06:49:15.474819  [**] [1:12799:3] SHELLCODE base64 x86 NOOP [**]
 [Classification: Executable Code was Detected]... {TCP} a.b.c.d:15017 -> W.X.Y.Z.:80
04/15-06:49:15.474819  [**] [1:12802:3] SHELLCODE base64 x86 NOOP [**]
 [Classification: Executable Code was Detected]... {TCP} a.b.c.d:15017 -> W.X.Y.Z.:80
04/15-06:49:15.619028  [**] [1:12800:3] SHELLCODE base64 x86 NOOP [**]
 [Classification: Executable Code was Detected]... {TCP} a.b.c.d:15017 -> W.X.Y.Z.:80
```

The injection was successful and Nepenthes captured and logged the malware submission:

```
[2011-04-15T06:49:19] a.b.c.d-> W.X.Y.Z. ftp://1:1@a.b.c.d:21/Rewetsr.exe
 c511c4f9bdd3bb892e582fbc9a00da9c
```

4 Case Study

This section details the UMD honeynet, the honeypot network deployed at the University of Maryland and also describes how DarkNOC is used to operate and maintain this particular network.

4.1 UMD Honeynet

4.1.1 Introduction

The honeypot network hosted at the University of Maryland was initially built in 2004 with unused IP addresses of the campus network. More recently, other organizations joined the initiative: AT&T Labs, the University of Illinois at Urbana Champaign, and the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) in Toulouse, France. Each of these organizations contributes to the UMD honeynet by providing ranges of public IP addresses.

The objective of the UMD honeynet is to provide the infrastructure to support honeypot-based experiments. The network features a centralized data collection and guarantees a realistic but controlled and flexible environment to safely deploy experiments. The advantages of the present architecture are multiple:

- A single gateway collects and stores the stores Snort events, flow data and network traffic, providing visibility across the full range of exposed networks.
- The experiments are easy to deploy without the need to create tunnels or to setup specific network configurations.
- The UMD honeynet is scalable, new organizations can join the project by providing range of IP addresses.

4.1.2 Architecture

Figure 11 shows the current architecture of the UMD honeynet and the different institutions involved in the project. A tunneling program called Honeymole[3] redi-

[3] http://www.honeynet.org.pt/index.php/HoneyMole

rects silently the traffic from the different organizations to the UMD honeynet.

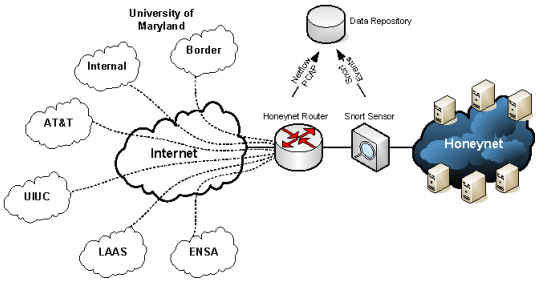

Figure 11: UMD honeynet architecture

The complexity of managing and monitoring such a network was the primary motivation for the development of DarkNOC. This section will discuss the application of the tool to that problem.

4.2 UMD DarkNOC Implementation

4.2.1 Subnet Status

The subnet status section is specific to the UMD honeynet. Each organization involved in the UMD honeynet provides one or more ranges of IP addresses called subnets. For example, the University of Maryland provides two distinct subnets: a subnet of the campus internal network and a subnet at the border network. The failure of a Honeymole tunnel is a significant event for the network, as it implies loss of an entire subnet; the subnet status display allows a manager to quickly assess the status of the tunnels and act on any issues.

Each subnet hosts a low interaction honeypot run by Nepenthes to collect malware. Depending on the network configuration, a Honeymole tunnel may be established to redirect the traffic to Maryland. DarkNOC monitors the quantity of malware collected, the status of the Honeymole tunnels, and the status of the low interaction honeypots.

4.2.2 Compromised Honeypots Detection

Some experiments deployed on the UMD honeynet may present significant risks. In the likely event of a honeypot being compromised, the attacker may use the machine to attack other hosts on the Internet. These attacks are generally easily detectable: Figure 12 shows that the volume of outgoing traffic is substantially greater than the incoming traffic. In this case, a honeypot was used as a proxy server.

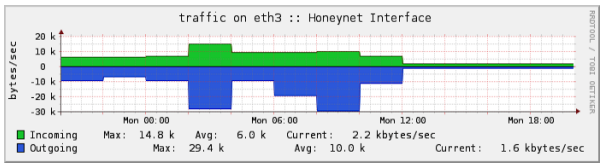

Figure 12: Network traffic (04/18/2011)

4.2.3 Traffic Anomaly Detection

A current experiment uses a known-vulnerable SSH server running on about 80 IP addresses of the Internet subnet provided by the University of Maryland. The DarkNOC's summaries proved useful in analyzing an attack on this configuration of the network which occurred on June 3, 2011.

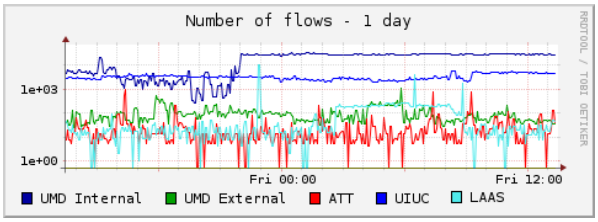

Figure 13: 06/03/2011, number of Flows

1. Figure 13 shows an increase in the number of flows just before midnight on Thursday night.
2. The number of attackers presented in Figure 14 remains relatively steady. This suggests that a fixed set of attackers is generating a large volume of traffic.
3. Figure 15 shows that port 22 is very active. As SSH sessions do not usually generate many flows, we can assume that the attacker is using a bruteforce attack against several IP addresses hosted within the UMD honeynet.
4. The word cloud of the honeypots targeted showed that the IP addresses of this specific SSH experiment were targeted.

DarkNOC provided several indications on the nature of the attack responsible for the spike in traffic network and flows. That night, the health monitoring system of the experiment reported several times that the machine was overloaded and the SSH server failed.

4.2.4 Using Honeypots as a Security Tool

Compromised Hosts Detection

The network traffic observed within an honeypot network is considered malicious. A healthy host would

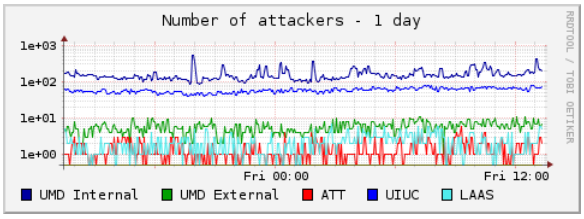

Figure 14: 06/03/2011, number of attackers

Top 10 Ports (last 24 hours)		
Port Number	**Number**	**%**
TCP / 22	240597	57.49%
TCP / 27977	20000	4.78%
UDP / 42332	18247	4.36%
TCP / 80	11248	2.69%
TCP / 1433	8464	2.02%
TCP / 443	5552	1.33%
UDP / 19756	5550	1.33%
TCP / 3306	5033	1.2%
TCP / 3389	4513	1.08%
ICMP / 8.0	4286	1.02%

Figure 15: 06/03/2011, top 10 destination ports

```
--------------- Analysis Report ---------------
Flow Time Window: 2011/06/06.06:00:00-2011/06/06.18:00:01
Number of hosts detected: 3
To access the online version of the report:
  https://xxx.xxx.xxx.xxx/darknoc/alert_hosts.php?report=263

xxx.xxx.xxx.xxx (X.umd.edu)
 - Number of flows: 1
 - Number of packets: 1
 - Number of bytes: 51
To visualize the flows:
   https://xxx.xxx.xxx.xxx/darknoc/alert_hosts.php?id=1124

yyy.yyy.yyy.yyy (Y.umd.edu)
 - Number of flows: 10
 - Number of packets: 10
 - Number of bytes: 1915
To visualize the flows:
  https://xxx.xxx.xxx.xxx/darknoc/alert_hosts.php?id=1125

zzz.zzz.zzz.zzz (Z.umd.edu)
 - Number of flows: 10
 - Number of packets: 10
 - Number of bytes: 1915
To visualize the flows:
  https://xxx.xxx.xxx.xxx/darknoc/alert_hosts.php?id=1126
```

Figure 16: Alerting module report

not normally communicate with the honeypots. We can therefore use the UMD honeynet to detect compromised hosts on the Maryland campus network. We assume that if a computer on campus appears in the flow data, that means the host is compomised. The alerting module queries the flow data to identify these hosts. This method is efficient at detecting scanners: the use of subnets from both local and remote sites means that a scanner is likely to eventually visit the UMD honeynet whether its probes are directed locally or at the Internet.

When a compromised machine is detected, the alerting module analyzes the event and generates an email that is sent to the IT Security Officer for further analysis. Figure 16 is an example of such a report. For each host, the number of flows, packets and bytes are provided. The report is also available on the Web interface of DarkNOC, it is possible to vizualize the flows associated with the alert. This technique helps to identify compromised hosts and misconfiguration as well. When this alerting system was first launched, the IT team figured that even if a host was tagged as blocked in their systems the compromised host was still able to communicate on the network and to continue its malicious activity. The analysis is performed every 12 hours and each participating organization gets notified of the eventual compromises of their systems. The choice of running the analysis at this frequency was chosen based on the feedback provided by the security team of the University of Maryland. The team wanted to receive a report early in the morning and right after business hours.

Security Profiling

Honeypots can provide relevant information regarding attackers and their techniques to compromise a computer. DarkNOC brings together enough information from different datasets to establish a security profile of a network. This profile includes the services targeted, the number of malware uploaded and the types of attacks. The objective is to help the security officers and network administrators to understand where to focus their efforts and to identify weaknesses and misconfigurations. DarkNOC can also be used to evaluate the performance of the security policy in place. The attacks detected and the malware uploaded on the honeypots are good indicators of the efficiency of an IPS device.

Attack techniques are constantly evolving as new vulnerabilities are discovered regularly. The honeypots can help to identify the current trends and to update the security policy accordingly.

5 Related Work

Lance Spitzner defines honeypots as a security tool *whose value lies in being probed, attacked, or compromised* [21]. In other words these are highly monitoring computer systems meant to attract hackers, analyze their modus operandi and profile them [19]. Placed in production environments, honeypots take an active part in the security of a network by providing information on attackers and attacks' patterns. Niels Provos introduces two types of honeypots [18]: high interaction honeypots

that involve the deployment of real operating systems on real or virtual machines, and low interaction honeypots that are computer software emulating operating systems and services.

Companies and researchers currently deploy honeypots networks at different scales. Also known as honeynets, these honeypots networks can be limited to few IP addresses on the local network or distributed systems in several locations such as the Leurre.com project [16], the Internet Motion Sensor [4], SGNET [13] or the honeynet initiative from CAIDA [23].

Levine et al. demonstrated the usefulness of deploying honeypots accross large enterprise networks [14]. In their study, Snort [20] was used to detect compromised computers accross Georgia Tech network. In DarkNOC a similar detection has been made possible by using the flow data. We assume that any traffic seen on the honeypot network is malicious.

The visualization and data analysis of malicious network activity has been the focus of a variety of commercial and open source products. On the commercial side, security companies such as Tenable and Sourcefire offer threat management products that collect logs from multiple devices and generate alerts to inform security analysts about potential intrusions. The main limitation of these solutions with respect to our goal is that they are not tailored to honeypot management and honeynet data collection and so they require additional effort to integrate honeypots in the organization security data analysis suite. Arbor Network is another commercial security vendor that offers a threat management product but the difference with the previous solutions is that they leverage their customer networks to instrument dark IP space at a large scale. As a result, they offer a global view of malicious network activity through their Atlas portal[4], which provides functionalities similar to DarkNOC, with graphs and tables for top attacks, top threat sources and attack trends.

On the open source side, the main honeynet management solution has been Honeywall [8] developed by the Honeynet Project. The Honeywall is a bootable CD-Rom that installs a Linux-based network gateway to manage and control honeypots as well as visualizing and analyzing honeynet logs. Compared to DarkNOC, Honeywall has a more capabilities to actively limit outgoing traffic but it has been designed for small honeypot network. The data processing capabilities of DarkNOC were designed for large scale and multi-site deployments. The objective of the DarkNOC project is to provide a flexible and powerful analysis program. It is adjustable to fit different honeypots configurations. However Honeywall is a all-in-one solution for small scale honeypot networks. It provides routing, capture and analysis capabilities. Integrating Honeywall in an existing large-scale honeypot network is more challenging.

Other open source projects that are not specifically tailored for honeypots include Alienvault [7], Aanval[5], Nfsight [6] and NVisionIP [12]. Alienvault and Aanval are network and system log management solutions that can only process Snort alerts and syslog events while Nfsight works exclusively with Netflow and has been designed for large-scale processing and security visualization of Netflow. NVisionIP processes global network Netflow data to specifically detect attacks and misuses.

Visoottiviseth et al. present a distributed honeypot framework using low interaction honeypots [22] running the honeyd daemon [17]. More specifically, they describe the working of the honeyd logs centralization and their analysis [22]. The framework only works with Honeyd log files. The level of interaction of our framework is also different since we are running low interaction honeypots as well as high interaction honeypots.

6 Future Work

We are working on a number of extensions and improvements on DarkNOC. The first extension will be the addition of a malware section in the user interface. This new section will provide more information about the malware collection including a graph showing the number of uploads per day but also some indications on the methods used to upload the malicious software and its name. The second improvement will be the implementation of the automatic detection of compromised honeypots in the alerting module. This detection will allow DarkNOC to automatically block the outbound traffic of compromised honeypots. Currently, only the detection of compromised non-honeypot hosts of an organization is automated. The graphical user interface of DarkNOC can also be enhanced. There is no option that allows to select and display the activity of a specific period of the day. It would be useful to be able to choose on a graph a particular moment of the day and see the activity at this precise time.

7 Conclusion

In this paper we presented DarkNOC, a honeypot network management and monitoring tool. DarkNOC provides a summary of the activity of the honeypots in the network. This summary is generated from different sources of data including Netflow, malware collected by the Nepenthes low interaction honeypots and attacks detected by the Snort intrusion detection system. Brought

[4] http://atlas.arbor.net

[5] http://www.aanval.com/

together, these data sources provide important resources to help network administrators, security teams, and security researchers understand attacks and protect systems. DarkNOC can be used the detect traffic anomalies and identify interesting case study for research purposes. Since it is important to detect quickly any compromised honeypots in the honeynet, DarkNOC provides administrators of these networks information regarding the health of the systems. Security teams may find a particular interest in DarkNOC since it can be used to detect compromised honeypots as well as compromised hosts on their non-honeypots networks. To sum up an organization using DarkNOC can have a better understanding of:

- the most targeted systems,
- the attackers, the attacks and their origin,

but also, DarkNOC helps:

- to obtain an overview of Honeynets activity,
- to identify security tools and devices misconfiguration.

Acknowledgement

The authors thank the Office for Information Technology at the University of Maryland. In particular we thank Gerry Sneeringer and his team for allowing the deployment of the UMD honeynet, providing feedback on DarkNOC and investigating the compromises detected by the application.

References

[1] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann. Characterization of attackers' activities in honeypot traffic using principal component analysis. In *Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing*, pages 147–154, Washington, DC, USA, 2008. IEEE Computer Society.

[2] Paul Bacher, Thorsten Holz, Markus Kotter, and Georg Wicherski. Know Your Enemy: Tracking Botnets (using honeynets to learn more about bots). Technical report, The Honeynet Project, August 2008.

[3] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling. The nepenthes platform: An efficient approach to collect malware. In *Proceedings of RAID'2006*, pages 165–184, 2006.

[4] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The internet motion sensor: A distributed blackhole monitoring system. In *In Proceedings of Network and Distributed System Security Symposium (NDSS 05*, pages 167–179, 2005.

[5] R. Berthier and M. Cukier. An evaluation of connection characteristics for separating network attacks. *International Journal of Security and Networks*, 4:110–124, February 2009.

[6] R. Berthier, M. Cukier, M. Hiltunen, D. Kormann, G. Vesonder, and D. Sheleheda. Nfsight: netflow-based network awareness tool. In *Proceedings of the 24th USENIX LISA*, 2010.

[7] Jeramiah Bowling. Alienvault: the future of security information management. *Linux J.*, 2010, March 2010.

[8] G. Chamales. The honeywall cd-rom. *Security Privacy, IEEE*, 2(2):77 – 79, mar-apr 2004.

[9] Kevin Curran, Colman Morrissey, Colm Fagan, Colm Murphy, Brian O'Donnell, Gerry Fitzpatrick, and Stephen Condit. A year in the life of the irish honeynet: attacked, probed and bruised but still fighting. *Inf. Knowl. Syst. Manag.*, 4:201–213, December 2004.

[10] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 581–590, New York, NY, USA, 2006. ACM.

[11] Jan Kohlrausch. Experiences with the noah honeynet testbed to detect new internet worms. *IT Security Incident Management and IT Forensics, International Conference on*, 0:13–26, 2009.

[12] Kiran Lakkaraju, William Yurcik, and Adam J. Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, VizSEC/DMSEC '04, pages 65–72, New York, NY, USA, 2004. ACM.

[13] Corrado Leita and Marc Dacier. Sgnet: A worldwide deployable framework to support the analysis of malware threat models. In *Proceedings of the 2008 Seventh European Dependable Computing Conference*, pages 99–109, Washington, DC, USA, 2008. IEEE Computer Society.

[14] J. Levine, R. Labella, H. Owen, D. Contis, and B. Culver. The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks. In *Proceedings of the IEEE Workshop on Information Assurance*, IEEE Systems, Man and Cybernetics Society, pages 92–99, West Point, NY, June 2003.

[15] Mauro, Ro, and Francesca Mazzoni. HoneySpam: Honeypots Fighting Spam at the Source. pages 77–83.

[16] Fabien Pouget, Marc Dacier, and Van Hau Pham. Leurre.com: on the advantages of deploying a large scale distributed honeypot platform. In *ECCE'05, E-Crime and Computer Conference, 29-30th March 2005, Monaco*, 03 2005.

[17] Niels Provos. Honeyd: A Virtual Honeypot Daemon. Technical report, Center for Information Technology Integration, University of Michigan, February 2003.

[18] Niels Provos and Thorsten Holz. *Virtual honeypots: from botnet tracking to intrusion detection*. Addison-Wesley Professional, first edition, 2007.

[19] Daniel Ramsbrock, Robin Berthier, and Michel Cukier. Profiling attacker behavior following ssh compromises. In *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '07, pages 119–124, Washington, DC, USA, 2007. IEEE Computer Society.

[20] M. Roesch. Snort - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX LISA*, 1999.

[21] L. Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[22] V. Visoottiviseth, U. Jaralrungroj, E. Phoomrungraungsuk, and P. Kultanon. Distributed honeypot log management and visualization of attacker geographical distribution. In *Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on*, pages 23 –28, may 2011.

[23] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Scalability, fidelity, and containment in the potemkin virtual honeyfarm. *SIGOPS Oper. Syst. Rev.*, 39:148–162, October 2005.

[24] Nathalie Weiler. Honeypots for distributed denial of service attacks. In *Proceedings of the 11th IEEE International Workshops on Enabling Technologies: nfrastructure for Collaborative Enterprises*, pages 109–114, Washington, DC, USA, 2002. IEEE Computer Society.