Advanced Threat Analytics for Incident Response

GCIH Gold Certification

Author: Darren Spruell, phatbuckett@gmail.com
Adviser: Rick Wanner

Accepted: October 6th 2007

Table of Contents

Introduction

The coursework presented in the SANS GCIH program provides a simple and effective structure for security professionals to apply to incident handling. The approach outlines a commonly accepted set of steps that guide analysts through basic incident response procedures. This paper is written with the intent of providing security analysts and incident responders with expanded threat analysis methodologies that go beyond those covered in the GCIH coursework. By definition, incident response nearly always implies a reactive state of activity where action is being taken in reply to notable events that occur in the environment. Many security analysts spend much of their time engaged in a perpetually reactive cycle of receiving and responding to security events. Still, others find that they have both the time and desire to engage in proactive threat identification and mitigation. Trends in recent years have resulted in many threat vectors evidenced as activity that originates from inside the network and reaches outbound, providing justification for threat identification and the opportunity to seek out these threats on the internal network environment. Some ideas for sources of data found in a typical enterprise network environment that can be used to fuel this proactive threat identification are presented.

Incident handling and threat analysis

Incident handling is a term which describes a formalized process of identifying and responding to security incidents in a structured manner (SANS, 2006). Threat analysis is a concept most often associated with security threat intelligence, an area which focuses on gaining knowledge of new and existing threats for the purpose of formulating defenses to mitigate them. In the formalized incident handling process, threat analysis fits into the *preparation* and *identification* phases. The data gathered during threat analysis can then feed into the phases which come afterward, supplementing the entire incident handling process.

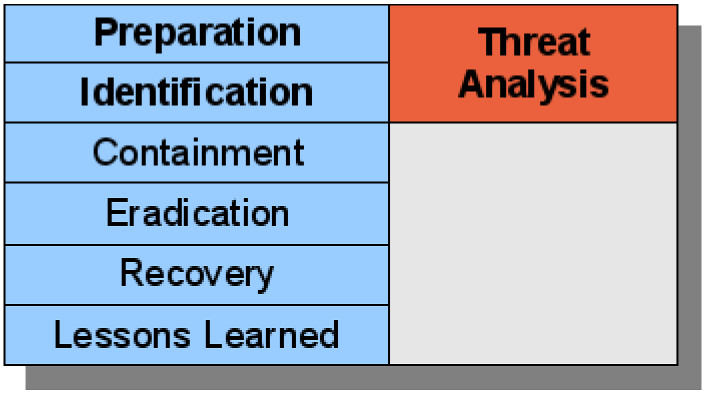

Illustration 1: Incident response phases

Preparation phase

Handlers who are not actively participating in later stages of working an incident should be spending time in the preparation stage. From a threat analysis standpoint, this stage is critical and can be leveraged for a couple of purposes. Firstly, this phase can serve as a lull during which handlers can take the time to learn about and research new and existing threats that may target the environment. This is the basis of threat intelligence and is key to providing information that can be used later to identify and respond to incidents arising from those threats. Secondly, this time should be used to put in place various tools, utilities and data sources that can be used in later phases. During the time spent researching threats, handlers should try to focus in part on what is currently missing from the available analysis resources that will needed to aid in identification of those threats. These resources might be in the form of credentials and login access to systems in the environment, access to host, network or application logs, or placement of intrusion detection systems. It might be relationships with key users or groups who can provide incident data originating in their respective areas. It might also be in the form of processes and procedures for any other type of data acquisition that can be managed.

Identification phase

From a threat analysis standpoint, handlers may consciously begin to rely on the tools, utilities and resources put in place during the preparation stage. Use and analysis of those resources provides a way of proactively identifying those threats in the environment and gathering the necessary information in order to respond to them. As handlers gain knowledge of new threats which may target their users and information systems, they may take the threat attributes they've learned about and begin to search through the tools and data sources at their disposal in an attempt to identify where those threats have penetrated defenses. By applying the precepts of the identification phase to these resources, it is possible for handlers to see where these threats may have made their way into the environment as well as where they have resulted in successful compromises.

Threat analytics for incident response

The above outline of steps in the early stages of the incident response process constitutes a framework that can be applied to many specific focal areas in the environment. If there is a system which can be leveraged for security threat data, the cycle of *prepare/analyze/identify/respond* (PAIR) can be used to more effectively leverage these systems and data with analytical techniques to help locate and eradicate threats in the environment.

The remainder of this paper focuses on several key areas that can be leveraged in this manner. Each focal area is presented in terms of the PAIR framework and provides direction for carrying out each part of the framework in order to use it effectively.

Proxy servers

The first systems explored are proxy servers, commonly used in many organizations to handle connections to Internet servers on behalf of clients. These devices sometimes provide content caching services and are frequently implemented to provide a centralized access control and policy enforcement gateway. The most common type of proxy servers in use are HTTP proxies, which operate at the application layer and can communicate with web, FTP, and other servers which speak HTTP and similar protocols. More capable proxies can speak other application protocols and transcend the web proxy role.

Because of their use as a centralized gateway for web access, proxies are often leveraged as a resource for Human Resources and Management when investigating corporate policy and conduct violations and employee Internet usage statistics. Not surprisingly however, proxy logs are also useful to incident handlers due to the amount of data contained in log entries which is helpful during analysis. Most proxy server software implements very detailed logging formats such as the W3C Common Log Format or native log formats which may provide more information. Fields you should expect to find in your web proxy logs include the following (W3C, 1995):

- request timestamp
- request URL
- status code
- bytes transferred
- client IP address
- destination server IP address
- request method
- authenticated user ID

This focus on log data forms the basis for the preparation and identification phases.

Preparation: The steps taken in this phase may differ depending on the organization and their use and management of proxy servers. Some organizations currently use proxy servers and rely on them as a critical infrastructure component, restricting outbound web access to only proxied clients and implementing access controls as a security measure. Others may use proxies opportunistically and provide them as an optional service in the environment which can be used to realize a performance increase in web browsing from the caching functions they provide. Some smaller shops may not implement proxy servers at all. In order to gain the greatest advantage in terms of threat analysis, proxies should be implemented as controls and policy gateways through which all outbound web access must flow. Doing so ensures that all web activity is monitored and recorded by the proxies, reducing or eliminating gaps in the ability to account for what actually occurs in terms of both user- and malware-initiated web access.

For organizations which currently rely on proxies in the environment, steps may need to be taken to provide access for the incident response team if the proxies are managed and administered by a different group. The key is to ensure that timely and full access to log data is available. Depending on the proxy implementation, this may be a matter of configuring the product to send a separate log stream (e.g. syslog feed, FTP batch transfer, etc.) to a system which can be monitored by the incident response team. If the product itself does not provide this capability, a script to copy files on a regular basis to an analysis host will often work.

Identification: The focus of the identification phase is using proxy logs to discover and validate security threat data as it may pertain to incidents or events in your environment. By forcing client web access through your proxy server(s), you can be sure that the proxy logs will reflect all intentional user browsing activity as well as all other web access activity which may not have been user initiated. Viruses, worms, bot and spyware agents, even those which go to great lengths to hide their presence and activity on end systems, can be quickly revealed this way. A useful application of the identification phase in connection with proxy logs is their use in uncovering client access attempts for known malicious content. This application is explored in detail next.

Identifying malicious content delivery

Proxy logs provide a wealth of information about the resources requested by clients on the internal network. Due to their standard and consistent logging formats, they can also be easily parsed and searched for strings and patterns.

A good way to start with identifying malicious content delivery is with searches for the most commonly used carrier for malicious files. On Windows platforms, this is frequently in the form of PE executables with a *.exe* extension. There are a good number of legitimate applications that will be downloaded that are distributed as EXE files. These are often distributed from well known domains/servers and are named in a manner that reflects the name of the program. Many others that do not fit these attributes are malware. During analysis of proxy logs, be on the lookout for indicators such as the following:

- Files downloaded from servers with no FQDN, only IP address
- Files downloaded from host-bouncing service domains or other dynamic DNS names known to be hostile, such as *3322.org*, *8800.org*, *8866.org*, etc. (F-Secure, 2006).
- File names that are short or random, such as *a.exe*, *1.exe*, or *4jx3yt12.exe*.
- Files from a single server or web directory following a numerical naming sequence, such as *81.exe*, *83.exe*, *84.exe*, *86.exe*.
- Files downloaded from domains masquerading as legitimate domains, usually with those domains as a subdomain of a hostile second level domain.
- Files downloaded from domains tied to typo squatting, where the domain name is a slight misspelling of a different, legitimate web site name.

As examples of the above, the following are a listing of URLs retrieved by clients that were confirmed to be known malware as observed in proxy logs:

No FQDN, sequential file names:

```
hxxp://209.11.244.34/Images/1.exe (PWS-LegMir.gen.h)
hxxp://209.11.244.34/Images/2.exe (PWS-LegMir.gen.h)
```

Sequential file names:

```
hxxp://xxx.axgzba3.com/x/97.exe (PWS-OnlineGames.z)
hxxp://xxx.axgzba3.com/x/98.exe (PWS-OnlineGames.r)
hxxp://xxx.axgzba3.com/x/99.exe (PWS-OnlineGames.s)
```

Masquerading domains:

```
hxxp://www.yahoo.americangreetings.com.dowsamk.net/macromedia-
flashplayerupdate.exe (Proxy-Agent.af)
```

A further approach along these lines is to fully integrate proxy log searches into your threat intelligence initiatives. By monitoring various security information resources which can provide information on current and emerging threats on the Internet, you can begin to respond to them proactively. Such resources include the DNS-BH project (DNS-BH, 2008), Sunbelt Blog (Sunbelt, 2008), Mal-Aware.org (Mal-Aware, 2008) and the ii (Incidents & Insights) Discussion group. For example, one trend in malware distribution has been the creation of fake media codec projects which distribute malware posing as video codec engines. Users will be directed to download and install these codecs from various other sites on the Internet. By means of threat intelligence gained from research and resources such as those mentioned above, a list of download sites known to host malicious fake codecs can be built. A query of proxy logs reveals several downloads from sites hosting these trojans, the full URL of the downloaded files, and the client IP address:

```
1189801373.409 1772 10.2.184.74 80 TCP_HIT/200 - codec-club.com HTTP/1.0 203572
486 DIRECT/64.28.184.178 application/octet-stream 64.28.184.178 -
ICAP_NOT_SCANNED GET none OBSERVED [user-agent stripped]
http://codec-club.com/download/codec-club4085.exe
1190053632.159 449 10.2.78.49 80 TCP_MISS/200 - codecplus.com HTTP/1.1 88860 419
DIRECT/codecplus.com application/octet-stream 64.28.184.179 - ICAP_NOT_SCANNED
GET none OBSERVED [user-agent stripped]
http://codecplus.com/download/codecplus1328.exe
[...]
```

Each of the files downloaded by clients was then retrieved on an analysis station and scanned using current antivirus software. Each of the downloaded files results in a successful detection:

Downloaded file	Malware detection
hxxp://proporno.org/codec/**codec.exe**	Generic.dx trojan
hxxp://**codec-club.com**/download/codec-club4085.exe	DNSChanger.ka
hxxp://**codecplus.com**/download/codecplus1328.exe	DNSChanger.jf
hxxp://**codecname.com**/download/codecname4413.exe	DNSChanger.ka
hxxp://**codecportal.com**/download/codecportal11370.exe	DNSChanger.jf
http://**codecmpg.com**/download/codecmpg4279.exe	Puper.gen.d

Based on this information, the incident handler can then verify installation and correct operation of antivirus software on each of the identified clients, review the logs for the software, and if needed, continue further in the incident response process to isolate and recover from any potential compromise that may have occurred.

Email servers

Email servers are the next item on the list which may be leveraged for threat data. Next to the web, email is likely the next most common malware delivery channel for end systems. One differentiator between web and email delivery is the relative level of awareness that has accompanied email-borne malware in recent years. Many organizations who accept email from the Internet also employ some form of content filtering solution on their gateway email servers. Blacklists and greylists are employed to filter out SPAM senders, and integrated antivirus programs scan inbound messages for viruses and other malware. The same is true for the major webmail services, as most of them will also scan email attachments for malicious content. A large amount of focus has been placed in modern organizations' user awareness training programs around the security issues surrounding reading of unsolicited emails and message attachments. Still, a number of threats do make their way into the enterprise via email -- even with the virus scanners, block lists, and other measures. Botnets and fast-flux DNS have replaced open relays and known SPAM servers with large networks of machines which bombard organizations with email. Content filters continue to be a step behind the more inventive spammers, as described in the SecureWorks writeup of a spam campaign which hit the Internet in October of 2007 (Stewart, 2007). Messages containing malicious attachments have given way to embedded links and social engineering ploys which lure users to attackers' sites, enticing them to give away confidential information and loading malware onto the systems by exploiting browser flaws and user ignorance. Regardless, threat intelligence can again assist in detection and identification of these threats which penetrate an organization through the email gateway.

Preparation: As with proxy servers, the key to threat identification and analysis with email is to have access to the mail server logs. Depending on the configuration and makeup of an organization's messaging infrastructure, different levels of mail servers may exist and various products may be in use. On a given mail server, logs may be found from any of the SMTP or other email services present, as well as any antivirus services, as well as any content filtering software. The incident handling team should be given access to these logs as each of them provide valuable information during threat identification.

A second step to take in the preparation phase is to gain access to resources which can provide early warning and intelligence on email-borne threats. Frequently these are in the form of directly attached malware, but often it will be phishing emails and other emails instructing users to visit a remote site where computers may be compromised via vulnerabilities in a web browser or other client-side software. Good resources include:

- AVIEN and AVIEWS
- Incidents & Insights Discussion Group
- SANS Internet Storm Center (ISC)
- SecureWorks/LURHQ

A second common approach to take with security threat data is networking with peers and other organizations in your industry. Many organizations in the same industry face similar threats and undergo many of the same attacks. At the highest level, every Internet-connected organization will face the same common set of threats. Arrange intelligence sharing initiatives with those who are willing to share information with you in exchange for details you can provide them. With these relationships in place, work out a set of procedures and protocols by which contacts will be initiated and documentation can be shared.

Identification: Again, the focus of the identification phase is using logs to discover and validate security threat data. After positioning your organization to receive information from the various intelligence sources, you may begin to parse the information you receive into actionable search terms. This application is explored in detail next.

Identifying malicious content delivery

Discovery of malicious content delivery via email depends on several circumstances that may exist depending on an organization's mail handling solution. Any of these strategies can drastically reduce the amount of content that makes it past the mail gateway and into user mailstores.

- Some organizations utilize aggressive mail filtering focused on policy blocks for specific file extensions, such as .exe, .pif, .scr, and .bat.
- Many organizations employ antivirus at the email gateway which will scan attachments and delete known malware.
- Many organizations utilize vendor-produced blacklists, DNSBLs, and score-based content filtering (such as Bayesian filters) to block messages believed to be from illegitimate senders.

In the end, the malicious content that makes it through any of these defenses does so because it works around the limitations of each of them. Malicious files can bypass policy blocks by being sent with an extension considered “safe”, by embedding them in .jpg, .gif, .doc, .pdf and similar files or compressing them into ZIP and RAR archives. Antivirus utilities may be evaded by utilizing new malware or variants for which signature detection is not available yet. Another tactic taken by the malware senders is to employ social engineering methods in the email message, enticing the end user to open an encrypted ZIP archive attached to the message which includes the password to the archive. Because the ZIP file is password-protected, gateway AV cannot scan it and will often let the attachment pass. Finally, Bayesian filters and other score-based systems have long been an area where spammers can nudge ahead in the cat-and-mouse game. Many of the SPAM consoles utilized by the senders allow a spammer to check the Bayes score prior to sending a message, allowing them to tune the contents to make it through these content-scoring systems (Stewart, 2007). Once you understand the limitations your defenses may have, begin analysis based on the threat information you've gathered.

- The two most important endpoints for analysis for email threats are the SMTP gateway (typically an MX host) where the message initially entered the environment, and the end-user systems where the email may be received.
- On the external-facing gateway server, important attributes to note are the date and time of the initial receipt, the delivering SMTP client, the envelope sender and recipient addresses, and where available, the contents of the Subject header. Note that the final Received header placed by your mail gateway is the only one that can be fully trusted, as past Received headers can be spoofed by intermediary mail relays.
- On the end system, the full contents of the received email message are needed for analysis. This includes the full RFC822 headers as well as the body of the message. This is needed to reconstruct the chain of SMTP servers through which the message was relayed en route to the recipient. Be aware that Received headers up until the point that a message is delivered to trusted SMTP servers may be subject to forgery to mask the true origin of a message.

- It is sometimes necessary to track a message through the entire chain of SMTP deliveries if your messaging infrastructure has many levels. For example, a given message may traverse a frontend MX host in the DMZ, a content scanning engine in the DMZ, one or more SMTP connectors internally, and final delivery to a mailbox server. At any point configurations could alter message delivery flow. When tracking delivery of email, this is a potentially important point.

Firewalls

Firewalls are now a de facto security measure implemented for network defense in most organizations. Firewalls typically provide several functions, operating as routers which interconnect networks, packet filters which enforce security policy at the IP level, and sometimes provide more advanced functionality in terms of traffic control and filtering. Most organizations employ at least one firewall at the network perimeter which segregates their network from the Internet. In this capacity, firewalls serve as a valuable resource from which threat analysis data can be taken.

Preparation: Several preparatory steps can be taken with respect to firewalls:

- **Access to logs:** Firewall logs are of value from a post-event, forensics standpoint. Typical options for logging allow ruleset creators to selectively log per firewall rule. Many organizations log on deny rules only. If possible, try to enable logging on all rules or as many rules as possible while staying within bounds of performance restrictions imposed by logging overhead. Doing so will ensure that the firewall logs can serve as a full audit trail whenever the firewall becomes a data source when an incident occurs. Incident handlers should have access to firewall log data in whichever fashion allows the logs to be easily queried and mined for information. Some firewall implementations allow users to display logs but do not provide useful filtering or searching capabilities. From an incident handing perspective, these capabilities are critical. If the firewall provides log access via logging consoles (such as Check Point's implementation), handlers should be given access to the console. If raw textual logs are available, those should be granted as well due to their usefulness in parsing and text processing.

- **Interactive traffic analysis:** Many firewall implementations provide the capability for administrators to view or capture network traffic interactively from a console. Unix-based firewalls which provide access via a shell interface (such as Check Point SecurePlatform, Linux netfilter/iptables, and platforms such as OpenBSD/PF) typically allow an administrator to capture or display traffic in real time using utilities such as *tcpdump*, *snoop* and *fw monitor*. Two possibilities exist in relation to incident responders' ability to benefit from interactive traffic analysis; either the handlers can be given direct access to the firewall interface to perform the analysis, or relationships and procedures should be put in place with the group responsible for firewall management to make sure that they can service requests from the incident handlers to perform packet captures on their behalf and in a timely manner.
- **Protocol and application information:** As network traffic is observed on the wire, a number of IP and application protocols will be found to be in use. This final item of preparation

concerns incident handlers' ability to uncover the meaning and purpose of the network traffic they observe crossing the firewalls. In this area of preparation, ensure that the incident handling team has the necessary relationships and contacts in place to find answers to questions about the traffic observed. A database of hosts or inventory of hosts and applications in use on the production network is very useful for this purpose. Especially when confronted with a proprietary protocol, information about the software that produced it and any possible protocol diagrams are useful in understanding an application's or system's role and purpose in relationship to an incident that is underway.

Identification: Utilize the firewall logs along with searches for host and/or destination IP addresses and protocols or ports to identify hosts involved. Attributes such as frequency or duration of communication and timing of repeated communication can be important indicators of notable activity. Look for logs evidencing communications to and from hosts known to be hostile (based on threat intelligence gained in preparation phase). Review firewall logs for signs of scanning activity on the network, often evidenced by high numbers of destination hosts and one-to-many communication. Finally, leverage the ability to take packet captures from the firewall when an incident has been declared to acquire full packet logs from the host(s) involved in the incident. This can be valuable for network forensic investigation and post-compromise analysis.

Identifying compromised hosts

Utilizing firewalls as a method for the identification of compromised hosts on a network is possible with the above preparation and identification concepts. A number of strategies exist, consisting of both post-event analysis using firewall logs as well as interactive analysis via real-time packet captures.

- Compromised hosts frequently scan for additional systems which provide services that may be vulnerable to compromise as well. Look through the logs for signs of one-to-many communication patterns over TCP and UDP services commonly targeted by worms. These include Windows NetBIOS and SMB/CIFS services (135/tcp, 139/tcp, 445/tcp) and ICMP pings for host discovery (echo-request). If you've learned of other worms or attacks that may behave similarly, you can find evidence of them as well. For example, 2001 saw the Code Red and Nimda worms which scanned local networks and the Internet looking for web servers (F-Secure, 2001). Evidence of their propagation attempts and similar scans are seen on port 80/tcp (eEye, 2001). Attackers utilize brute force and password guessing scripts to break into Unix systems with weak credentials via SSH and frequently install toolkits to scan for other systems. Their traffic can be identified as scans and sweeps on port 22/tcp. Be careful to note systems which may exhibit this behavior as part of normal operation, however. DNS servers frequently communicate with many hosts on the Internet over port 53/udp and 53/tcp. The same goes for hosts running email services which may connect to many hosts on the Internet via SMTP (25/tcp). On the other hand, be on the lookout for unauthorized SMTP speakers attempting to communicate outbound on port 25, as this is a good indicator of an infected host attempting to propagate via SMTP or delivering keylog data to an attacker.
- If your threat intelligence has uncovered information about

known malicious or hostile hosts on the Internet, search your logs or perform ad hoc traffic captures looking for communication to or from these hosts. Attackers frequently base attacks from and host malicious software on many of the same servers such as those utilized by the Russian Business Network (RBN); hosts found communicating or attempting to communicate with servers such as these can typically be found to have been compromised by trojans or other malware relying on these known malicious hosts. A useful technique to employ for interactive analysis at this point is to build a filter input file containing a filter for hostile IP addresses, which can be specified with the -*F* switch to tcpdump(8). A similar collection of entries may be compiled into a pattern list file that can be utilized by grep(1) with the -*f* switch, which is useful for searching through textual firewall logs.

- Search specifically for communication identifying bot command and control (C&C) channels. A large number of C&Cs rely on IRC for their master-drone communication, and many of these utilize the standard IRC ports of 6660-6669 and 7000 (ShadowServer, 2007). The ability to run IRC services over non-standard ports means that better analysis can be done at a level other than static port attributes. Organizations with users who regularly chat over IRC will find this analysis somewhat challenging, but it may be possible to enumerate the servers they connect to and exclude them from analysis.

Honeypots

Honeypots represent another technology that is incredibly useful in and of itself for the purpose of threat intelligence. Honeypots are designed as computer resources whose only purpose is to get exploited by attackers (Shadowserver, 2007). Honeypots are an important part of leveraging threat intelligence for incident response as they are one of the technologies most useful for producing usable threat intelligence data directly. In reality, much of the threat data posted to the public from independent security research projects on the Internet comes from honeypots, including the Honeynet Research Alliance and the mwcollect Malware Collection Alliance as examples. Organizations wishing to leverage honeypots in their arsenal of network security tools and tactics are encouraged to research their use and available configurations and consider any legal and privacy issues with their use.

Preparation: Key considerations for leveraging honeypots include the type(s) of honeypot desired and the capabilities they provide. Honeypots can exist as single systems with one IP address and can scale into honeynets, collections of honeypots which may be configured with many IP addresses. Another form of honeypots focuses on network address space in quantities of several class B or class C sized networks and is more focused on the makeup of traffic seen rather than what activity is carried out on honeypot systems. Whatever approach is taken by the organization, placement and reporting stand out as key issues. Deploy one or more honeypots in locations and according to configurations which will be most visible and attractive to potential attackers. While externally-facing honeypots will help shed light on the kind of attacks your organization is facing from Internet attackers, an arguably more valuable deployment tactic for the purpose of this paper is one or more internally-facing deployments, which can reveal threats that exist or originate inside the perimeter. After deploying the honeypots and configuring them in a manner that fits the intended use and threat data desired, formalize methodologies and procedures for reviewing collected data and feeding gathered intelligence into other network defense systems such as IDS and firewalls. Finally, since your organization will likely uncover malware- and human-driven attacks against your honeypots, put together validation and escalation procedures to assist with responding to those findings. You are likely to need relationships with leadership, human resources, legal, and other technology groups to complete these preparation steps.

Identification: Utilize the available monitoring and reporting functions that are available with your chosen honeypot implementations. At a minimum this will include logs produced by the software, and these may be simply reviewed regularly as with any system reports. Alternatively, automated analysis and response may be configured by utilizing scripts which can monitor the log files and employ pattern matching to trigger notifications or other event responses. Note that honeypots are typically deployed such that any network communications to the honeypot hosts is suspect and will indicate malicious or illegitimate activity. Attempt to identify internal hosts which are attempting to access resources on your honeypots and ascertain the reason for this activity. It may be that the hosts are compromised and an automated tool or an attacker is now moving on to more systems on your network. It may be that an insider has attempted to locate sensitive information and is probing your honeypot.

Darknets/sink holes

Darknets and *sink holes* are similar to honeypots in that they represent resources on a network to which no legitimate network traffic should ever be destined. A Darknet is a portion of routed, allocated IP space in which no active services or servers reside. Sink holes are similar in nature, often deployed at ISPs and major backbone providers. These are "dark" because there is, seemingly, nothing within these networks (Cymru, 2004). In reality, a system is configured to listen passively for network traffic which enters the darknet. This system records details about the network activity and source of the activity and allows security personnel to perform analysis and respond to the traffic. Like honeypots, traffic seen entering a darknet or sink hole can be considered anomalous and is very likely malicious in nature. These systems can provide a very effective method of gaining threat intelligence on attacker activity, sources of attack, and a sample of overall network traffic on the whole, doing so with a relatively low false positive rate.

Preparation: As with honeypots and honeynets, the required preparation for darknets and sink holes surrounds deployment. Initial decisions should be made as to the approach an organization wishes to take with these technologies. For the purposes of threat identification on the internal network, an internally-facing darknet should be set up. Address space must be allocated for use in the darknet; typically, one or more full class subnets (such as a class C (/24) or class B (/16) network) will be allocated, although organizations may wish to choose something smaller in size; in reality, one or more individual host addresses (/32) can be used, although by premise more is better. This IP space is then routed either via an internal gateway routing protocol (IGP) or static routes to the system which will function as the darknet or sink hole sniffer host. The incident response team with typically need to work with the group responsible for network management in the organization to accomplish this. The sniffer host which now becomes the target of this routed darknet traffic will then need to be configured with utilities which make analysis and recording of the received traffic possible. For a Unix-based system, utilities such as the ubiquitous tcpdump and Argus (http://www.qosient.com/argus/) and some network accounting packages such as SNMP and MRTG (http://oss.oetiker.ch/mrtg/) will provide much of the needed analysis and reporting capabilities.

Identification: The data provided by a darknet should be considered a fairly reliable indicator of what it represents; with the exception of misconfigured systems and applications, most of the traffic found destined to a darknet should be considered malicious. Using the utilities put in place on the darknet sniffer host, identify from the ports and protocols of the observed traffic what the activity represents. Much of what strays into the dark IP space will be traffic from agents scanning network space on the lookout for new hosts. Virus and worm outbreaks can be quickly identified by this scanning activity and attempted propagation patterns show up. Host IP addresses are easily available for identification and remediation of compromised systems.

Intrusion detection systems (IDS)

Intrusion detection systems are now one of the most ubiquitous forms of network attack alerting. IDS as a technology has had ups and downs in terms of viewpoints in the industry; analyst firm Gartner declared IDS to be “obsolete as of 2005” (Gartner, 2003), and yet many organizations still rely on and endorse IDS systems as a necessary component in their incident response arsenal. IDS as a market-driven technology underwent a fork to providing attack blocking capabilities with the advent of intrusion prevention systems (IPS) which have gained popularity in recent years. Nonetheless, traditional IDS are still a mainstay in the toolkit of many security analysts and incident responders. While flawed and prone to false positives, signature-based network IDS (NIDS) provide valuable capabilities with their attack detection and identification capabilities and are even recommended components of methodologies which focus on more holistic traffic flow analysis techniques. From the standpoint of threat analysis for incident response, we find that IDS can provide a critical supplemental role in the incident response and threat analysis processes, independent of whatever primary role it may fill in traditional intrusion detection at the core of a security program.

Preparation: As with most of the previously mentioned technologies, the key concept in preparation is placement of IDS sensors. Our primary focus in this paper is network IDS (NIDS), which can provide visibility into network traffic and associated attacks which are carried in that traffic. Deployment strategies at a detailed level are outside of the scope of this document, but placement in a position that can at least see egress of traffic from internal network segments is useful. Oftentimes network ingress and egress points are the same set of devices by design; a single firewall deployment can see flows inbound from and outbound to the Internet. Positioning of IDS sensors on similarly key network segments provides flexible and effective visibility. Equally important to placement and visibility is the set of IDS signatures which are loaded on an IDS system. Traditional IDS rules focus on detection of exploit attempts against known vulnerabilities in services and systems. IDS are typically designed to be flexible enough that they can be used in creative ways for detection of even non-traditional threats by modeling the rules to detect other attack situations. As an example, the Bleeding Threats (formerly Bleeding Snort) IDS rules collection contains a number of rules which transcend the traditional exploitation rules collection by addressing a number of current threats including spyware, web site SQL injection, viruses, botnet C&C hosts, top attacker IPs, and more (Emerging Threats, 2008). These and other rule sources can provide useful extensions to your threat detection capabilities. Select and install the rules that you wish to leverage for the profile of threats you wish to be alerted to. Pay attention to other sources of threat data which frequently produce detection signatures, such as can be seen in the SecureWorks analysis of the Gozi trojan (Jackson, 2007) which provided updated Snort IDS signatures for the threat. Utilize those signatures which will help you increase your capabilities for threat detection.

Identification: As NIDS events trigger alerts, identification of those threats comes with review and analysis of the generated alerts. NIDS typically provide very detailed data from all layers of the network communication from the frame level to the application payload that triggered the alert. Utilize this alert data to review generated event data and identify threats that you have uncovered. Better NIDS implementations will also generate full packet captures for the network traffic that results in the generated alert. This can sometimes provide more details when analyzed and is useful for loading the event communication into external applications such as Wireshark for protocol decoding and deeper investigation. Finally, be aware that an common criticism of signature-based IDS is the level of false positives that may accompany the valid threat alerts. With this in mind, understand that not every alert you see will be an attack or threat indicator – particularly when your IDS system is running some of the more experimental or bleeding-edge detection signatures that are stretching the rule language for increased detection capabilities.

Using NIDS to support threat analytics

One of the most useful applications of NIDS technology is in the supplemental role it can play during threat analysis and incident response. In a situation where network traffic captures are available, particularly in a large data set, a very quick litmus test to determine malicious activity in the capture to replay the traffic through a Snort instance configured with a "full" signature ruleset. This approach is advocated by Richard Bejtlich in his methodology for *Structured Traffic Analysis* where he proposes, "Snort can generate alert data as a fast way to identify low-hanging security fruit. Suspicious traffic identified by Snort can be examined using session data or full content data." (Bejtlich, 2005).

Just as firewalls and their logs were suggested earlier as analysis sources to use for identification of network scanning and malware propagation activity, NIDS also work well. Similar to firewalls, NIDS are often deployed at network egress choke points and have a very opportune view into traffic exiting the enterprise. This provides a useful vantage point for handlers to run packet traces and either capture or interactively review communication on the wire. For example, a useful tcpdump(8) filter to use for identification of the NetBIOS and SMB/CIFS scanning that can occur with common network worms can be used on NIDS sensors monitoring affected network segments:

```
tcpdump -n 'port 135 or port 139 or port 445 and tcp[13] = 0x02'
```

This filter displays TCP packets on the common ports targeted and with the SYN flag set, attempting to initiate a connection to other hosts. Run on a NIDS sensor, this can quickly show an analyst which hosts on the network are attempting to connect to numerous other hosts, often providing an easy indicator of compromised systems and facilitating identification of this activity. Manual packet inspection of this type is not necessary; indeed a number of approaches exist for identifying this scanning activity using IDS signatures, including the following rules from the Emerging Threats repository (Emerging Threats, 2008):

```
alert tcp $HOME_NET any -> any 445 (msg: "BLEEDING-EDGE Behavioral Unusual Port
  445 traffic, Potential Scan or Infection"; flags: S,12; threshold: type both,
  track by_src, count 70 , seconds 60; classtype: misc-activity; sid: 2001569;
  rev:11; )
alert tcp $HOME_NET any -> any 139 (msg: "BLEEDING-EDGE Behavioral Unusual Port
  139 traffic, Potential Scan or Infection"; flags: S,12; threshold: type both,
  track by_src, count 70 , seconds 60; classtype: misc-activity; sid: 2001579;
  rev:11; )
```

Similar rules can identify the outbound scanning from an SSH brute force scan or an FTP brute force attack:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 22 (msg: "BLEEDING-EDGE Potential SSH
  Scan OUTBOUND"; flags: S; flowbits: set,ssh.brute.attempt; threshold: type
  threshold, track by_src, count 5, seconds 120; classtype: attempted-recon;
  reference:url,en.wikipedia.org/wiki/Brute_force_attack; sid: 2003068; rev:2;)
```

```
alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (msg:"BLEEDING-EDGE SCAN Potential
  FTP Brute-Force attempt"; flow:from_server,established; dsize:<65;
  content:"530 "; depth:4; pcre:"/530\s+(Login|User|Failed|Not)/smi";
  classtype:unsuccessful-user; threshold: type threshold, track by_dst, count 5,
  seconds 300; sid:2002383; rev:8;)
```

IDS can be leveraged in many different ways and despite their shortcomings can be very useful as supplemental tools for incident response and threat analysis.

References

SANS. (2006). Security 504 GCIH Training Coursework. *Incident Response*.

W3C. (1995). The Common Logfile Format. Retrieved 3/3/08 from http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format

F-Secure. (2006). 3322, 8866 and others. Retrieved 3/3/08 from http://www.f-secure.com/weblog/archives/00000883.html

DNS-BH. (2008). Malware List. Retrieved 3/3/08 from http://malwaredomains.com/

Sunbelt. (2008). *Sunbelt Blog*. Retrieved 3/3/08 from http://sunbeltblog.blogspot.com

Incidents & Insights (2008). *ii Discussion Group*. Retrieved 3/3/08 from

http://npogroups.org/lists/info/ii

Mal-Aware.org. (2008). Mal-Aware.org mailing list. Retrieved 3/3/08 from

http://mal-aware.org/

Stewart, J. December 4, 2007. SecureWorks. Inside the "Ron Paul" Spam Botnet. Retrieved 12/29/07 from

http://www.secureworks.com/research/threats/ronpaul/

Jackson, Don. March 21, 2007. SecureWorks. Gozi Trojan. Retrieved 12/30/07 from

http://www.secureworks.com/research/threats/gozi/

F-Secure. (2001). Description of the Nimda worm. Retrieved 3/3/08 from

http://www.f-secure.com/v-descs/nimda.shtml

eEye. (July 17, 2001). Code Red Worm Analysis. Retrieved 3/3/08 from http://research.eeye.com/html/advisories/published/AL20010717.html

Shadowserver. (2008). IRC C&C Port Utilization. Retrieved 3/3/08 from http://www.shadowserver.org/wiki/pmwiki.php?n=Stats.IRCPorts

Shadowserver. (2007). Honeypot Information. Retrieved 3/3/08 from http://www.shadowserver.org/wiki/pmwiki.php?n=Information.Honeypots

Cymru. (2008). Darknets. Retrieved 3/3/08 from http://www.cymru.com/Darknet/

Gartner, Inc. (2003). IDS as a market failure. Retrieved 3/3/08 from http://www.gartner.com/5_about/press_releases/pr11june2003c.jsp

Bejtlich, R. (2005). Structured Traffic Analysis. http://www.net-security.org/dl/insecure/INSECURE-Mag-4.pdf

Emerging Threats Rules. (2008). Retrieved 3/3/08 from http://www.emergingthreats.net/rules/