Heat-seeking Honeypots: Design and Experience

Abstract
Compromised Web servers are used to perform many malicious activities. Attackers are constantly searching for vulnerable servers as they provide free resources and have high page ranks. In this work, the authors aim to understand how attackers find, compromise and misuse vulnerable servers. The authors present heat-seeking honeypots that actively attract attackers, dynamically generate and deploy honeypot pages. Then they analyze the logs to identify attack patterns. Over a period of three months their honeypots attract more than 44,000 attackers from 6,000 different IP addresses.

Introduction
Compromised web servers are commonly used for serving phishing and malware pages, redirecting user traffic to malicious sites. Almost 90% of Web attacks take place through legitimate sites that have been compromised. Over of 50% of popular search keywords have at least one malicious link to a compromised site in the top search result page.
A honeypot is a computer security mechanism set to detect, deflect, or, counteract attempts to gain unauthorized access to information systems.
• **Client-based honeypots**: are used to detect malicious servers that attack clients
• **Server-based honeypots** (such as heat-seeking): emulate vulnerable services or software and passively wait for attackers

The architecture of a heat-seeking honeypot

Steps:
1. Attacker do queries
2. Queries issue to search engines
3. The pages from the search results are fetched
4. Then these pages are advertised and crawled by search engines
5. When attackers issue similar queries to search engines, the honeypot pages are returned in the search results
6. Attackers interact with the honeypot,
7. all interactions are logged for further analysis.
Heat-seeking honeypot design have four components:

1. **Identify which Web services types, targeted by attackers (obtaining attacker queries)**
   Attackers often use two methods to find vulnerable Web servers. The first is to perform brute-force port scanning on the Internet, and the second is to make use of search engines to identify malicious queries.

2. **Dynamically generate and deploy honeypot pages**
   For each query obtained by the previous step is issued to Bing and Google search engines. Then the top three results are picked to be emulated. A crawler fetches those URLs and other elements required, such as images and stylesheets. All JavaScript content is stripped and all links are rewritten to point to the local versions. The HTML content is then encapsulated in a PHP script which logs all information regarding each visit to a database. The URL structure of the original pages is designed to be the same in the emulated ones. Moreover, they manually install a few common Web applications that are frequently targeted in separate VMs, so that if one of them gets compromised, it does not affect the working of the other applications.

3. **Advertise honeypot pages to attackers via search engines**
   Ideally, they want their honeypot pages to appear in the top results of all malicious searches, but not normal user searches. So they submit the URLs of the honeypot pages to the search engines and wait for the crawlers to visit them. Then, they increase the chance of honeypot pages (PageRank), by adding hidden links (not visible to regular users) pointing to the honeypot pages on other public Web sites.

4. **Analyze the honeypot logs to distinguish attacks from normal users and crawlers**
   Heat-seeking honeypots receive three types of traffic. The legitimate users traffic, the legitimate search engines crawler traffic and the malicious traffic.
   - **Identify Known crawlers**
     One way of identifying crawlers is from their agent strings. For example, the Google crawler uses Mozilla/5.0 (compatible;Googlebot/2.1;+http://www.google.com/bot.html) as its user agent. Moreover, they check if the IP address of the crawler should also match the organization it claims to be from. Finally, they identify crawlers by their access patterns (crawlers typically follow links in the page and access most of the connected pages).
   - **Identify Unknown crawlers**
     There are two types of links that a crawler can visit, static and dynamic links. A static link can be visited many times but a dynamic only once. By analyzing the Bing logs, regular expressions matching the groups of dynamic links are generated. The group of dynamic links is called E. Links that don’t match these sets are the crawlable group C. A crawler should access a large fraction of the URLs (P) with a **threshold K = |P|/|C|** and outside of group C, only pages from group E.
   - **Identify Malicious traffic**
     The static web pages on the heat-seeking honeypots are just used to attract attacker visits. But, they observe that most of the attackers are not targeting these static pages. The attackers try to access non-existent files or files that were not mean to be public accessed. Additionally, they create a whitelist with all the dynamic and static links, for each site. They considered a request to be from attacker if:
       a) It tries to fetch a link which isn’t in the whitelist, i.e., a link not accessed by a crawler
       b) The requested link should not be present on the server, i.e., request results in an HTTP 404 Error
Results
During three months of operation, the system consisted of 96 automatically generated honeypot Web pages (3 pages each from 32 malicious queries), and four manually installed Web application software packages. They altogether received 54,477 visits from 6,438 distinct IP addresses.

They analyzed the log files and they found that:

- Around the half of the software honeypot pages that contain dynamic links have been crawled by 3 search engines (Google, Yahoo and Bing), all the other links are visited by exactly one crawler.
- 16 ASes (comprising 375 different IP addresses) crawling more than 75% of the hosted pages, and another 18 ASes (comprising 79 IP addresses) visiting less than 25% of the pages. Therefore 75% is picked as the threshold K, and anyone visiting more than this is considered a crawler, while others are considered legitimate users that reach the honeypot pages.
- The most popular honeypot page with over 10,000 visits was for a site running Joomla, a content management system.
- The 10% of the IPs are very aggressive and account for over half of the page visits.
- Over 24% of the IPs are located in the US
- In the median case, the discovery time between the first crawl of the page by a search crawler and the first visit to the page by an attacker, was 12 days

They use three different honeypot setups:

1. **Web Server:**
   - The machine has no Hostname, so the only way to access it is by its IP address
   - There are no hyperlinks pointing to the server, so it is not in the index of any crawler/search engine

2. **Vulnerable software:**
   - Commonly targeted Web applications
   - The application pages are accessible on the Internet (there are links to the on public websites)
   - Search engines can find them

3. **Heat-seeking honeypot generated pages:**
   - Simple HTML pages that emulate vulnerable pages
   - Search engines can find them, too

- They show that the Heat-seeking setup is the most effective (more visits and attackers). Moreover, it received instances of almost all the different attack types.
- The attackers continuously scanning IP ranges looking for vulnerable web servers (they observed the first probe less than 4 hours after setting up the Web server)
- For 25% of the sites, almost 90% of the traffic came from attackers; i.e., less than 10% of the traffic was due to legitimate visitors. This happens mostly in the case of small or unpopular Web sites that do not have many regular visitors

Conclusion
The heat-seeking honeypots deploy honeypot pages corresponding to vulnerable software in order to attract attackers. They generate honeypot pages automatically without understanding the latest exploits or knowing which software is affected. The system can detect malicious IP addresses solely through their Web access patterns, with a false-negative rate of at most 1%. Heat-seeking honeypots are low-cost, scalable tools and that help us to understand attackers’ targets and methods, and that their use can effectively inform appropriate monitoring and defenses.