Chapter 17

USING SEARCH ENGINES TO ACQUIRE NETWORK FORENSIC EVIDENCE

Robert McGrew and Rayford Vaughn

Abstract Search engine APIs can be used very effectively to automate the surreptitious gathering of information about network assets. This paper describes GooSweep, a tool that uses the Google API to automate the search for references to individual IP addresses in a target network. GooSweep is a promising investigative tool. It can assist network forensic investigators in gathering information about individual computers such as referral logs, guest books, spam blacklists, and instructions for logging into servers. GooSweep also provides valuable intelligence about a suspect's Internet activities, including browsing habits and communications in web-based forums.

Keywords: Network forensics, search engines, evidence gathering

1. Introduction

Individuals and groups involved in penetration testing of network assets often use search engines to locate target websites. The search results may reveal information about unsecured administrative interfaces to the websites, vulnerable versions of web applications and the locations of these applications. Similarly, attackers seeking to deface websites or host phishing websites often attempt to identify targets with older versions of web applications with known vulnerabilities. Large numbers of potentially vulnerable hosts can be enumerated quickly using search engines. The Google Hacking Database [5] posts the results of such searches and provides applications that run the searches on target websites [5].

Information about computing assets collected by search engines can also be used in network forensic investigations. This paper describes the design and implementation of GooSweep, a tool for gathering network forensic information by performing searches using specific ranges

of IP addresses and their corresponding host names. Written in Python, GooSweep uses the Google Search Engine API to gather information about target networks without requiring direct communication with the networks [6]. In particular, GooSweep can provide useful network forensic information related to server compromise and web use policy violations. While the quality and quantity of information obtained by GooSweep may vary dramatically from one case to another, its ability to gather potentially valuable forensic information quickly and efficiently makes it a powerful tool for network forensic investigations.

The next section discusses how search engines can be used to obtain information pertaining to hosts and applications. Section 2 describes the GooSweep tool and its application to network forensics. The final section, Section 3, presents our conclusions.

2. Searching for Hosts

An Internet search for references to a specific IP address often returns instructions that inform users how to log into the host. For example, if an organization has a database server, there may be instructions on the organization's web server for employees, informing users about client software for connecting to the server, as well as the IP address and/or host name of the server. If an email server is among the hosts searched, its presence and purpose will often be apparent in the results, especially if users of the server post information to publicly-archived mailing lists. If these mailing list archives are accessible to the public via the web and indexed by search engines, emails to the list from users will often include detailed header information. The header information may contain the host name and IP address of the originating email server (allowing it to be indexed and discovered using the technique described in this paper), along with detailed version information about the email server software, client software and time stamps. Email message content provides useful information as well—one of our searches returned a post by a systems administrator seeking help with a specific software package.

Client workstations also provide varying amounts of information that may be indexed by search engines. Some web servers, including many government and academic systems, maintain access logs that are publicly accessible. This information can be used by a forensic investigator to identify the sites visited by users. In some cases, these log files include time stamps, operating system and web browser version information, and the referring URLs (the websites that led users to the destination) [1]. The referrals may also cite other websites that the users visited or reveal the search terms they used to arrive at the site that logged them.

Communications channels such as Internet Relay Chat (IRC), web-based forums and website guest books also record and display IP and host name information that may be indexed by search engines. When a user joins an IRC channel (analogous to a chat room) on most IRC networks, a line similar to the following is displayed to other users:

```
11:41 -!- handle [n=username@c-xx-xx-xx-xx.hsd1.mi.example.net]
      has joined \#channelname
```

In this example, `handle` is the name adopted by the user in the channel, `username` is the user name on the single-user workstation or multi-user system, and text following the `@` symbol is the host name of the computer that connected to the IRC server [7]. Often, users who frequent IRC channels will post logs of interesting chat sessions on the web. In the case of many open source projects, where meetings are held over IRC, all chat sessions on project channels are automatically logged and are publicly accessible. Search engines index all this information, enabling it to be found by tools like GooSweep. Web-based forums and guest books work in a similar way, logging and, sometimes, displaying the IP address or host name of the user who made the post in an effort to discourage spam and abuse.

Security-related information can also be found regarding hosts in a subnet. Spam blacklists, which contain lists of hosts known to relay spam email, are used by system administrators to track and block unwanted email; by design they contain host names and IP addresses [3]. If a system was once compromised and used as a platform for attacks or to host phishing sites, often there will be discussion on public mailing lists about blocking the machine or shutting down the host. This information is valuable to a forensic investigator as historical information about hosts or networks, or as intelligence about hosts and networks that were involved in an attack.

Querying Internet search engines for information about individual hosts in a range of IP addresses is promising because of the type of results it can return. In addition to facilitating network intelligence and penetration testing activities, the information gathered can be very valuable in incident response and forensic investigations.

3. GooSweep

GooSweep is a Python script that automates web searches of IP address ranges and their corresponding host names. Like many other search engines, Google does not permit automated scripts to use its normal web interface—these scripts increase the load on the interface and ignore advertisements. However, Google provides an API for programmers to de-

velop applications that utilize its search engine [2]. This enables Google to provide a separate interface for scripted search requests and also to limit the rate at which automated searches are conducted.

GooSweep uses the Google Search Engine API to perform searches. The API currently limits each script user to 1,000 requests in a 24-hour period. GooSweep uses a single API request for each IP address and each host name. With reverse-DNS resolution of host names enabled, an investigator can use GooSweep to search a class C subnet in a 24-hour period (256 hosts, each with an IP address and host name search, requires a total of 512 API requests). Fortunately, many networks do not have host names assigned to every IP address in their address ranges; this reduces the number of API requests required to scan a network. Consequently, an investigator can typically run GooSweep on two class C subnets in a 24-hour period. The "burst mode" can be employed for larger IP address ranges. This mode causes a script to idle after its API requests are expended; the script is activated when more requests can be issued. GooSweep generates an HTML report with the search results, including the number of websites found that match each host in the IP address range.

3.1 Running GooSweep

Executing GooSweep requires a Python interpreter [8] and the PyGoogle interface to the Google Search Engine API [4]. PyGoogle requires the SOAPpy web service library to be installed as well [9]. A GooSweep user must register for a Google API key to run scripts that issue queries. This key must be placed in a location specified by the PyGoogle documentation (typically in a file named `.googlekey` in the user's home directory). The GooSweep script itself is contained in a file named `goosweep.py`, which does not require any separate installation procedures. GooSweep has been extensively tested on Linux systems. Several users have had success running it on Windows systems without modification.

GooSweep may be executed from the command line using the following syntax:

```
./goosweep.py [-h num] [-r] [-b num] <[-d filename]
      | [-o report ]>  <-s subnet>
```

The required `-s` argument specifies the subnet to be searched. The argument is specified in "dotted-quad" format, with an asterisk as a wild card to denote the part of the address that is to be changed for each search. For example, `-s 192.168.5.*` directs GooSweep to scan the IP address range `192.168.5.0` through `192.168.5.255`.

Either or both of the `-o` and `-d` arguments are required to produce an output. A filename should be supplied to `-o` to produce an HTML report with horizontal bars indicating the relative number of hits for each host. A filename should be supplied to the `-d` option to generate a comma-delimited output file for analysis using other programs, e.g., Microsoft Excel,

The `-b` option, if specified, supports the burst mode. The Google API limits each user to 1,000 API requests in a 24-hour period. The burst mode option enables a user to specify the number of searches that GooSweep should perform in a 24-hour period. After performing the specified number of searches, GooSweep idles for the remainder of the 24-hour period and then continues with another set of searches. This allows GooSweep to automatically perform large scans without violating the limitations imposed by the Google API. Users may also use the `-b` option to budget the number of GooSweep API requests per day so that other Google API applications can run simultaneously.

The `-h` option enables the user to specify how often GooSweep should output hash marks (`#`) to the screen to indicate the progress of its search. The option may be turned off if GooSweep is being run as part of a wrapper script or application, or the option may be set as necessary to determine if GooSweep is running at a normal pace. The default option outputs one hash mark for every eight hosts searched.

The `-r` option allows the user to specify that a reverse-DNS lookup should be performed for each IP address, and if a host name is returned, that it is to be searched for as well. This option is turned off by default.

GooSweep was originally designed to provide information about a target network in a stealthy manner, without sending any packets to the target. A reverse-DNS lookup submits a DNS request to the target network, assuming that the result is not cached in a local DNS server. Issuing a large number of DNS requests can set off intrusion detection system sensors (these requests are often submitted by attackers performing network enumeration or reconnaissance). The `-r` option should be turned off during penetration testing in order to "fly under the radar." In general, reverse-DNS lookups should be activated in network forensic scenarios that involve scanning one's own networks.

The following is a sample GooSweep scan and dialog:

```
./goosweep.py -s 192.168.5.* -o report.html -r -h 4
#######...###
Generating report (report.html)
Completed.
```

For privacy reasons, the subnet scanned is in the "private" non-routable range. A report generated by the scan consists of an HTML

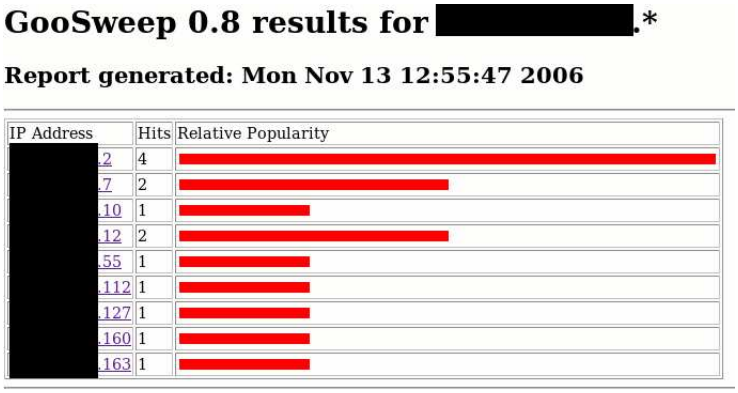

Figure 1. Sample GooSweep report.

table with each row containing an IP address, a host name (if the `-r` option is specified and a name is found), the results returned for each host, and a bar chart showing the number of hits for each host relative to other hosts in the scan. To assist digital forensic investigators, the IP addresses and host names are rendered as hyperlinks to the relevant Google search engine results.

3.2 GooSweep Example

Figure 1 illustrates the results of executing GooSweep, targeting a network typically used by students. The results have been censored to obscure the actual IP addresses scanned. IP addresses in the range that resulted in no search engine hits are omitted for brevity.

For each result with one or more hits, the IP address can be selected to view the corresponding Google search results. For most of the IP addresses in the example, web server logs were found at other academic institutions that had logged visits by these hosts. The visits were to web pages related to topics such as programming assistance and upcoming conferences. One IP address resulted in finding a security-related paper published at an academic conference that used a host in the address range in an example. The results dated as far back as 2004 and as recent as the current year (2006). Note that while this example was executed without reverse-DNS lookups, some of the web server logs contained the results of their own reverse-DNS lookups, allowing the naming scheme for this IP address range to be determined without having to issue queries using GooSweep.

4. Conclusions

GooSweep leverages the latest Internet search engine technology to provide valuable information gathering capabilities for network forensic investigators. There is no guarantee that GooSweep will be fruitful in any given situation, but few, if any, forensic techniques or tools can make this claim. Nevertheless, given its ease of execution and the richness of the information it can gather, GooSweep is an attractive tool for network forensic investigations. GooSweep and its source code [6] are available free-of-charge to members of the information assurance and digital forensics community.

References

[1] Apache Software Foundation, Apache Common Log Format (httpd .apache.org/docs/1.3/logs.html#common), 2006.

[2] Google, Google APIs (code.google.com/apis.html).

[3] N. Krawetz, Anti-spam solutions and security (www.securityfocus .com/infocus/1763), 2004.

[4] B. Landers, PyGoogle: A Python interface to the Google API (py-google.sourceforge.net).

[5] J. Long, The Google Hacking Database (johnny.ihackstuff.com/gh db.php).

[6] R. McGrew, GooSweep, McGrew Security Services and Research (www.mcgrewsecurity.com/projects/goosweep), 2006.

[7] J. Oikarinen and D. Reed, RFC 1459: Internet Relay Chat Protocol, IETF Network Working Group (www.ietf.org/rfc/rfc1459.txt?number=1459), 1993.

[8] Python Software Foundation, Python programming language (py thon.org).

[9] G. Warnes and C. Blunck, Python web services (pywebsvcs.source forge.net).