Divide and Conquer
Real World Distributed Port Scanning

Ofer Maor
CTO
Hacktics

16 Feb 2006 – Hackers & Threats I, 3:25PM (HT1-302)
• “Divide and Conquer: Real World Distributed Port Scanning” reviews conventional port scanning and distributed port scanning techniques, suggesting new approaches for implementing distributed port scanning in virtually any environment.

• After a short overview of port scanning we will examine the reasoning behind port scanning prevention and mechanisms which are used for such preventions.

• We will then proceed to reviewing existing distributed port scanning techniques, and their potential pitfalls, followed by a new approach for conducting distributed port scanning.
• This presentation is all about live demonstrations – discussed topics will be explained through actual execution of probes and scans.

• Today’s Objectives:
  — *Prove that conventional port scanning prevention does not really prevent port scanning, but rather creates a false sense of security*
  — *Provide security professionals with a new technique and a tool to conduct distributed port scans*

• The new technique as well as the tool presented today are the result of research done at Hacktics’ research labs, focusing on developing new techniques for identifying potential vulnerabilities.
Port Scanning Overview

- Port Scanning is the process of identifying some or all open ports (listening services) on one or more hosts.

- Usually conducted using automated tools, which can quickly scan through the entire potential port range (0-65535) of a server or search for selected common or expected ports.

- It is a crucial part of the attack’s reconnaissance phase, and is one of the first tasks done by an attacker when attempting to break into a machine.

- Free and commercial port scanners are widely available, varying in supported features, user interface and environment. The NMap scanner is probably the most common scanner in use.
Port Scanning in general, and NMap specifically, are so popular that even the creators of Matrix used it...
Port Scanning Overview

- Port Scanning is normally done by going over all (or some) TCP and/or UDP ports and checking the status of each port by attempting to communicate with it.

- With TCP scanning, three potential responses are available:
  - If an entire TCP handshake has been completed – the port is open.
  - If the server refuses the connection – the port is closed.
  - If no response at all is received (or if the firewall rejected the request) – the port is filtered.

- Similarly, half-open scanners that examine only the second phase of the handshake (SYN/ACK or RST/ACK) can be used for stealthier scanning, as can different scanning techniques based on various TCP/IP stack anomalies.
Port Scanning Prevention

• With the growing popularity of port scanning, many organizations felt the need to create protection against it (rather than solving the true problem at hand).

• This functionality is provided by various Intrusion Detection (or Prevention) Systems, known as IDS/IPS. These systems are capable of identifying multiple connection requests from a single host and block the IP address of this host.

• Most modern IDS/IPSs, including open source solutions such as Snort®, are capable of providing a working mechanism to protect against conventional port scanning.
In order to overcome port scanning protection, the next step in the evolution of the attack was the distributed approach.

The idea behind Distributed Port Scanning is to split the effort of the port scanning, having each host probe only few ports, thus not getting blocked by the system.

The gap between theory and practice, however, lies in the ability to control hundreds or thousands of hosts to perform this attack for the hacker or for the auditor.

Real hackers do not consider this a real problem, as they can gain access to a zombie network which can be controlled for a variety of purposes, including such scanning.
Divide and Conquer: Distributed Port Scanning

• Studies from last years show that large zombie networks can be purchased for 2,000-3,000$.

• Smaller networks are available for 500$ as well.

• Tools such as DScan and Unicorn can then be used for deploying the port scanning over the network.
The Auditor’s Problem

• Security professionals, such as auditors and pentesters, cannot rely on zombies to do the work, as running such a zombie network is highly unethical, as well as illegal in most countries.

• As a result, most penetration tests and audits fail to work around port scanning protection mechanisms, providing the customers with a false sense of security.
A workaround to this problem appeared in the form of the FTP Bounce Attack.

This attack allows taking advantage of a feature of the FTP protocol, allowing a client connecting to an FTP server to instruct it to connect to a third machine on an arbitrary port.

This behavior provided an opportunity for a wide variety of attacks, including the execution of distributed port scanning.

Modern FTP servers are no longer vulnerable to FTP Bounce attacks (and most firewalls block them as well). While this is generally positive, it unfortunately prevents security experts from using this behavior to conducting distributed port scanning.
• A similar workaround to this problem appeared later on in the form of the Idle scan.

• This type of scan takes advantage of the common behavior of IP stack implementation, which increment an identifier known as the “fragment identification” by one for every packet they send.

• It is therefore possible to spoof a packet supposedly originating from a 3rd party host, then examining the fragment identification of that host (if the port answered back to the host, this number will be incremented).

• Like with FTP Bounce, however, most modern operating systems have fixed this behavior, and any existing firewalls will not allow performing such enumeration in the first place.
Finding a Widely Available Resource

• The solution to distributed port scanning must therefore present itself using a resource complying to the following criteria:
  — Widely Available – To ensure that a large network can be easily created, evading the most sensitive port scanning protection.
  — Free – To guarantee that anyone wishing to perform such an assessment is capable of doing so.
  — Legal – Does not require the auditor to perform hostile acts against hosts on the internet.

• Luckily, such resources are available for use on the internet – Free HTTP Proxies.
Free HTTP Proxies

LISTS OF FREE PROXY SERVERS

Here you can find the lists of free proxy servers. You can get HTTP proxies and SOCKS proxies lists.

We have in our proxy database:

- HTTP proxy servers
- anonymous HTTP proxy servers
- high anonymous HTTP proxy servers (elite HTTP proxy servers)
- HTTPS proxy servers
- SOCKS 4 / SOCKS 5 proxy servers

Our proxy database statistics:

<table>
<thead>
<tr>
<th></th>
<th>HTTP²</th>
<th>HTTPS²</th>
<th>SOCKS 4 / 5²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of proxies per day (on average)</td>
<td>2000 - 10 000</td>
<td>100 - 500</td>
<td>200 - 1000</td>
</tr>
<tr>
<td>Number of proxies per week (on average)</td>
<td>10 000 - 20 000</td>
<td>1000 - 4000</td>
<td>1000 - 5000</td>
</tr>
<tr>
<td>Total proxies in base</td>
<td>300 000</td>
<td>50 000</td>
<td>150 000</td>
</tr>
</tbody>
</table>
Real World Distributed Port Scanning

- Using HTTP Proxies, it is therefore possible to achieve a real world solution for the auditor problem, allowing easy execution of distributed port scanning.

- The HTTP proxy based scanning can be achieved in one of two main techniques:
  - Via HTTP CONNECT requests – Easy, but rare.
  - Via HTTP GET requests - By utilizing HTTP Proxy Response Fingerprinting.
HTTP CONNECT Scan

• The CONNECT method is part of the HTTP RFC and is designed to allow tunneling of traffic which can not be normally proxied, such as HTTPS data.

• The usage of CONNECT as a port scanner is trivial – The proxy is contacted, and a CONNECT request is being sent to the proxy, requesting it to connect to the probed port.

• If the port is available, the following response is presented, identifying it as an open port:

```
HTTP/1.0 200 Connection established
Proxy-agent: Apache/1.3.12 (Unix) (Red Hat/Linux) mod_ssl/2.6.6 OpenSSL/0.9.5a PHP/4.0.1p12
```
HTTP CONNECT Scan

• Alternatively, the proxy will return a “Connection Refused” or “Gateway Timeout” error to indicate that the port was not available.

• By using CONNECT probing it is therefore possible to identify the 3 possible states of the port: Opened, Closed or Filtered.

• While the HTTP Proxy Port Scanner Tool supports HTTP CONNECT requests, this type of scan can not be relied upon, due to two key factors
  — Many free proxies will not support the CONNECT method at all
  — Even those that do support it, will usually only allow connecting to specific ports, and will therefore be of little value for our cause.
The HTTP method supported by virtually all free proxies is GET.

While GET requests are not aimed at performing a TCP connect at the remote host, they do however connect as part of the process, and can also be used to connect to non standard ports by providing the port number after the host name, such as http://host:1234/.

By relying on this functionality, it is possible to effectively instruct the proxy to connect to remote hosts using arbitrary ports.

While the response is not clear as with the CONNECT requests, it is possible to determine the port status by fingerprinting the response code, headers and body.
Like other fingerprinting mechanisms, such as OS fingerprinting or HTTP Server fingerprinting, the basic idea was to map out the behavior of common proxy servers available on the internet when attempting to connect to various types of open, closed and filtered ports.

After analyzing the list of given proxies, it is possible to run the same tests against scanned machines, identifying open ports, while distributing the test between the thousands of different free proxies.

Following is a sample behavior analysis of the popular Squid proxy.
HTTP Proxy Response Fingerprinting

Response I:

HTTP/1.0 200 OK
Location: SomeSite
Content-Type: text/html
Server: Microsoft-IIS
Content-Length: 214
Date: Mon, 13 Jun 2005 23:07:58 GMT

...
Response II:

+OK Welcome to MailEnable POP3 Server
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command
-ERR Unknown command

Identification: Tested port is open and runs a POP3 Server. Most other non HTTP open ports will behave similarly.
Response III:

```
220 domain.com ESMTP MailEnable Service, Version: 1.73-- ready at 04/20/05 21:25:32
503 Bad sequence of commands
503 Bad sequence of commands
503 Bad sequence of commands
503 Bad sequence of commands
503 Bad sequence of commands
503 Bad sequence of commands
```

Identification: Another normal open port was identified. Note that the 503 string should not be confused for a proxy response code.
Identification: This indicates that the port is closed. The body provides even additional supporting data:
While trying to retrieve the URL:

http://MyHost:1053/

The following error was encountered:

- Connection Failed

The system returned:

(111) Connection refused

The remote host or network may be down. Please try the request again.

Your cache administrator is root.

...
HTTP/1.0 504 Gateway Time-out
Server: Squid/2.3.STABLE1
Mime-Version: 1.0
Date: Mon, 13 Jun 2005 22:34:31 GMT
Content-Type: text/html
Content-Length: 697
Expires: Mon, 13 Jun 2005 22:34:31 GMT
X-Squid-Error: ERR_CONNECT_FAIL 110
X-Cache: MISS from proxy
Proxy-Connection: close
...

Identification: This indicates that the port is filtered. Similarly, additional supporting data can be extracted from the body.
As can be easily seen from these cases, Squid’s behavior is easy to fingerprint in order to use it as a port scanner:

— When the data returned is a valid HTTP response with 503 as its response code, the port is **Closed**.

— When the data returned is a valid HTTP response with 504 as its response code, the port is **Filtered**. (Note that unlike with NMap scanning, filtered only applies to dropped requests)

— When the data returned is not an HTTP response, or is an HTTP response with a different code, the port is **Open**.

Similarly, other common proxies can be fingerprinted to identify different responses according to the status of probed ports.
Fingerprinting Pitfalls

• The main problem when performing such probing are limitations presented by the proxy itself.
  — Some proxies allow port 80 only
  — Some proxies allow port 80 and high ports only
  — Some proxies allow only other specific ports

• In such case the proxy returns an error or simply ignores the request.

• Additionally, some proxies may not have as distinct fingerprints as Squid does, making differentiation between closed and filtered ports harder.
And some more proxy (pit)falls…

Proxy Falls, Willamette National Forest, Oregon
• Moving from theory to practice – Hacktics’ Distributed Proxy Port Scanning Tool

• The tool allows auditors and security professionals to perform a distributed port scan, through automation of the entire process:
  — Identifying free available HTTP proxies
  — Checking whether identified proxies can be used for port scanning
  — Identifying the relevant fingerprints of proxies in the list
  — Executing a port scan against a list of all (or selected) TCP ports, through distribution of requests between all proxies

• Tool Demonstration…
Additional Potential Use

• Using a similar framework, it is possible to:
  — Perform distributed vulnerability scanning, defeating IDS IP-based aggregation
  — Distribute web application attacks, thwarting off all IP-based protection and aggregation
  — Test for weaknesses against distributed denial of service attacks

• Future Release: Proxy Distributor Tool – A proxy utility running on the local computer, which distributes all HTTP activity.
Conclusion

- Using IDS to prevent port scanning provides false sense of security against focused attacks. Similarly, all aggregated IP based blocking can easily be overcome by distributed means.

- Effective protection must come in the form of a hardened environment, which does not rely on obscurity for achieving security and will not be put under a threat by a mere port scan.

- Moreover, performing IP based blocking allows hackers to easily cause denial of service attacks of commercial proxies.

- Using the new techniques presented, identifying weaknesses to distributed attacks is no longer the domain of malicious hackers, and can now be conducted by security professionals.
Questions?
Additional Information

The detailed paper which this presentation was based on, as well as the tool presented during the presentation can be found at:

http://www.hacktics.com/resources.html

For additional information or discussion of these topics, feel free to contact us:

info@hacktics.com
• NMap Tool & Information: http://www.insecure.org/nmap/

• USA Today Zombie Network Article: http://www.usatoday.com/tech/news/computersecurity/2004-09-08-zombieprice_x.htm
