Who Is Scanning the Internet?

Roman Trapickin
Betreuer: Oliver Gasser, Johannes Naab
Innovative Internet-Technologien und Mobilkommunikation SS2015
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: roman.trapickin@tum.de

ABSTRACT

Today's port scanning software is able to perform massive port scans up to the complete IPv4 space within minutes. Once a device is connected to the Internet, it will be almost immediately scanned for open ports and services. Most of these scans are done by anonymous individuals, in particular targeting at finding and exploiting system vulnerabilities. However, there is also a variety of individuals and organizations openly practicing massive port scanning and pursuing different objectives. In this paper we will introduce several scanning entities, while emphasizing their motives. In addition we will propose an entity classification model. In order to endorse further understanding of the topic we will also discuss port scanning as a measurement discipline as well as introduce the contemporary port scanning software used for Internet-wide scans.

Keywords

port scan, internet-wide scan, measurement, entities, nmap, zmap, masscan, classification, caida, sonar, sipscan, conficker, shodan, hacienda, legality, ethics

1. INTRODUCTION

IPv4 address space consists of 2^{32} or almost 4.3 billion possible IP addresses, which is within the *giga* order of magnitude. Today's computers have CPUs with gigahertz clock rates and gigabytes of RAM and storage. Today's networks allow bandwidths exceeding 1 gigabit per second. This makes iterations over the entire IPv4 space possible within a comparatively short time period. By comparison, IPv6 address space consists of 2^{128} or nearly 340 undecillion (10^{36}) addresses, which is rather unlikely to iterate over entirely in the nearest future.

Since the massive transition to IPv6 has not yet begun and IPv4 is still by far the dominant Internet protocol,[18] we now have a unique opportunity to reach every IPv4 address on the Internet in reasonable time. The year 2013 has seen ZMap and masscan emerging, two port scanning tools, which promised to port scan the entire Internet in under an hour on consumer hardware. Particularly ZMap has received an extensive IT media coverage,[2][10][31] which once again raised the discussion about port scanning and associated threats.

In this paper we will attempt to answer the question "Who is scanning the Internet?" by giving an insight into the current trends of massive port scanning. First, we will briefly introduce the technical terms used for describing port scanning activities in section 2. Popular software tools will be discussed in section 3. In section 4 we will talk about the individuals and organizations, which openly perform Internet-wide scans. We will focus on understanding their intentions, used tools and techniques. Finally, we will endeavor a classification model for these entities in section 5. As a short supplement we will revitalize the discussion about the legal and ethical aspects of (massive) port scanning in section 6.

2. SCANNING BASICS

The term "Internet(-wide) scan(ning)" describes a *port scanning* procedure performed by a single or multiple scanning entities on a considerable amount of hosts. There is no clear requirement or specification at which point scanning multiple hosts may be dubbed an "Internet scan", however scanning the complete IP spaces of regional Internet registries or even countries definitely falls into this term. In this section we will shortly describe the main reason behind such massive port scanning initiatives and also port scans in general as well as introduce main concepts and terms used throughout this paper. Although rudimentarily explained in this section, the real motives will be introduced in sections 4 and 5.

The main purpose of port scanning is *gathering information* about offered services from hosts connected to a network. This is done via sending probe messages to targeted hosts and prompting a response later on. As the name suggests, port scanning is centered around Transport layer ports and hence mostly based on the Transmission Control Protocol (TCP). Nevertheless both adjoining Network and Application layers often provide additional information about the targeted system.

2.1 Port Status

First, we explain what kind of knowledge a scanning entity can receive from a TCP segment. The essential step in (Internet-wide) port scanning is to determine the port status of a remote host:[24]

open port indicates that an application is accepting connections on this port. Finding an open port is the main goal for a scanning entity, since it discloses the most knowledge about targeted host.

closed port indicates that there is no active application on the other end. Bearing significantly less information about the host, a closed port could still provide some clues about e. g. the operating system by fingerprinting the TCP/IP stack. Various operating systems exhibit char-

doi: 10.2313/NET-2015-09-1_11

acteristic behavior while generating the freely selectable TCP/IP fields, so that collecting multiple responses and matching them against a database with known fingerprints makes OS detection possible.

undetermined port status is the least desirable message for a scanning entity. The port scanning software does not receive any response. In most cases the port is **filtered** – protected by a firewall. Some port scanning software, most prominently nmap, which offers various scanning techniques that – under favorable circumstances – may return more informative results. As a matter of consequence, the entire scanning process slows down drastically. The scanning techniques will be introduced in section 3.1.

In addition to Transport and Application layer knowledge, a scanning entity can acquire relevant information from the Network layer. Querying a WHOIS database with targeted IP address will provide the scanning entity with approximate host geolocation and ISP data among other things. A variety of WHOIS services is publicly available on the Internet. In general, retrieving a short description of running services is called **banner grabbing**.

2.2 Defining the Scope of a Scan

The next step is to define the scope of Internet scanning. This scope is defined by two dimensions, namely ports and hosts. A naïve calculation for IPv4 address space for every port results in $2^{32} \cdot 2^{16} = 2^{48}$ communication endpoints to be scanned. However, subtracting private IP ranges as well as blacklisting some IP addresses will not change the order of magnitude. By comparison, brute-forcing a 48-bit cryptographic key space is considered feasible today,[6] but still only on dedicated hardware.[29] Even though iterating over 2^{48} combinations alone can take significant amount of time on consumer-grade machines, actually it is the *bandwidth* that creates the major bottleneck. While scanning a remote host on the Internet, the scanning performance depends on bandwidth of every path segment a packet has to traverse. In fact, there are various factors, such as congestion control, which negatively affect the overall bandwidth. As a result, such scans are indeed possible, albeit rather impractical. Instead, scanning entities concentrate their effort on relevant ports, so that time to completion can be thoroughly improved. There exist three major approaches to reduce the scanning scope:[22]

Horizontal scan describes a port scan performed for the same port on multiple hosts. An extreme example of a horizontal scan is a **/0 scan** (entire IPv4 space) on a single port. Horizontal scans are often used by attackers to detect open ports on a large number of machines in order to exploit vulnerabilities of the listening application. In turn, such scans can be used for massive security audits, i. e. measuring global distribution of a vulnerability.

Vertical scan describes scanning multiple ports on a single host. Scanning every port out of 2^{16} of a host is called **vanilla scan**, while scanning a small subset is dubbed **strobe scan**. Such scans are mostly done for vulnerability detection on single systems. Obviously, such limited scope is not suited for Internet-wide scans.

Block scan is a combination of both horizontal and vertical scan, therefore a scan of multiple ports on several hosts. An extreme example is a /0 vanilla scan. As mentioned before, large block scans are poorly scalable, since including a single host to the target domain results in up to 2^{16} additional ports. In practice only a small number of ports per host is scanned. Hence an Internet-wide block scan may rather be considered as a series of few horizontal scans or alternatively as a /0 strobe scan. Such scans are as well interesting for attackers aiming at multiple vulnerabilities and security experts doing global research, but also useful for various service discovery tasks.

Figure 1 visualizes the three types of scanning in terms of targeted scope.

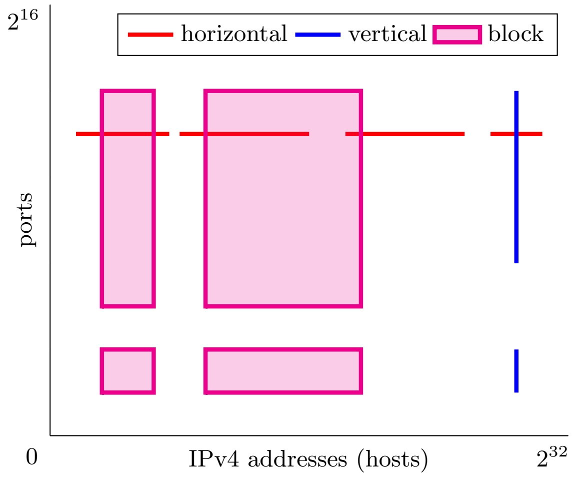

Figure 1: A simplified visualization of scan types

2.3 Measurement Standards

Finally, we will discuss port scanning from a scientific, or more precisely, experimental point of view. Regardless of the real motivation of a scanning entity, a (massive) port scanning initiative is a *quantitative research*, and, as such, it is subject to several measurement and quality standards. Vern Paxson from the International Computer Science Institute in Berkeley, California has proposed the following aspects for sound Internet measurement:[25]

Precision describes a degree of relative proximity between individual measured values. A telling example is the sampling rate of a continuous signal, whereas Paxson cites an example of clocks with 1 µs and 1 ms precision each as well as *filtering* responses. He also states, that, concerning the Internet measurement, precision is "readily apparent" and hence is rather of secondary importance. However, according to Paxson, a *sound* measurement study should at least respond to precision concerns.

Metadata is the data that accompanies the actually measured data. A prominent example of metadata *preservation* are (human-readable) logs which save additional information during the measurement. Metadata is an important aspect for Internet-wide scans, since it is possible to extract crucial information about port status and listening application from accompanying data.

Accuracy is a degree of relative proximity of measured values to the reference value. Acquiring such value can be done by comparison with values from other sources or previous measurements. When talking about Internet-wide scans, accuracy is often used to describe an effort spent for checking port status. An accurate scan therefore exhausts every possibility to get the best result when a simple check returns an undetermined status. As already mentioned before, elaborate scanning techniques may drastically decrease performance. Most purpose-built scanning tools (e. g. ZMap, masscan) often sacrifice accuracy for the sake of performance.

Misconception refers to false interpretation of results based on faulty measurement execution. A misconception commonly occurs as a consequence of leaving a critical detail out of account. For example, considering host as powered off, though it is actually hidden behind a firewall is a misconception.

Calibration is process of applying techniques for efficiently exposing inaccuracy and misconception issues. This includes i. a dealing with outliers and spikes or comparing multiple measurements. These techniques will not be further discussed in this paper.

Aside from the above aspects, which are supposed to be considered in order to avoid observational errors during the measurement, Paxson also introduces the best practices for working with (read: analyzing) measurement data:

Large volumes of data may lead to unexpected behavior of analysis tools, in turn slowing down the research process drastically. Paxson states, that extracting multiple data samples and analyzing them individually as well as comparing them for common properties and differences, will help to gain an overview about the data before proceeding with analyzing the whole dataset.

Reproducibility of analysis is a fundamental principle, found in every experimental scientific field. Reproducibility describes the ability of a third party to re-enact the experiment for the sake of process validation and verification of results. It is on behalf of the researcher not only to describe the experiment in detail, but also to provide a comprehensible “master script” for compiling the whole analysis chain, thus making its result quickly reproducible.

Public availability of data supplements the reproducibility of analysis. However, publicly available research data not only adds value to comprehensibility of the research, but also contributes to a “common framework”, which may be used by different researchers to confirm their results. Paxson endorses the publication of detailed datasets (including metadata), but also addresses the problem of disclosing sensitive information.

2.4 Scanned Entities

A port scan is a bilateral activity, and, as such it affects both scanning and scanned entities. As the last part of the Scanning Basics, we will shortly introduce the possibilities of detecting scans from third parties.

Because of the nature of port scanning, one cannot tell with 100 % certainty, whether an incoming connection is a port scan without confirmation from the scanning entity. Several “friendly” entities, while scanning HTTP services, tend to put contact information into metadata, e. g. short description and a web link in the HTTP User-Agent field that can be quickly seen by a system administrator. However, without this information detecting a port scan is far less obvious. A lot of heuristic detection methods have published, spanning from evaluating intensity of connection establishments[22] to probabilistic models.[21]

Both legal status and ethical aspects of port scanning have been a controversial topic. Opponents consider port scans as a service misuse, mainly because of resources spent on establishing and maintaining connection. Besides, some service providers perceive such thorough examination as inappropriate from ethical standpoint. Legal status and ethical issues will be discussed in section 6.

3. SCANNING SOFTWARE

With rising popularity of the Internet in the mid-1990’s, the first publicly available port scanning software emerged. Nmap was one of the first port scanners and has outlived most of its – meanwhile discontinued – competitors such as scanrand and unicornscan. Virtually every port scanner is capable of scanning IP address and port ranges (read: *block scans*), however scanners with limited scan scope also exist, for instance the OS X built-in Network Utility is only capable of vertical scans. Over time, various port scanners have introduced different probe request and response handling paradigms. For example, scanrand was among the first that worked with two processes for sending and retrieving responses asynchronously.[3]

In the following we will introduce the most renowned port scanning software, namely Nmap, as well as newcomers ZMap and masscan, which have drawn much attention promising to complete an Internet-wide scan within minutes. We will also use the opportunity to explain the different approaches used by these port scanners regarding the probe handling and dispatching.

3.1 Nmap

Nmap is the best-known port scanning command-line utility. It was specifically designed for both massive scans as well as scanning single hosts. The original author, Lyon “Fyodor” Gordon, has been developing Nmap since 1997, and with increasing functional complexity, development was later overtaken by the user community. Over its fairly long period of existence, Nmap has become almost synonymous with port scanning. Besides port scanning it has an extensive set of features, including host discovery, detection of running services and operating systems, as well as scripting for automation purposes. Nmap is also highly tunable through a variety of command-line parameters.[24]

Nmap excels in offering diverse scanning techniques, being dubbed a Swiss Army knife of port scanning for that reason:[24]

TCP SYN scan is “the default and most popular scan option”. Nmap sends a SYN message awaiting a SYN|ACK for an open or RST for a closed port. Lack of response indicates, that the port status cannot be determined.

TCP connect scan utilizes `connect()` – an operating system call – instead of crafting own messages. As opposed to TCP SYN scan, complete connection is established thus exposing the scan to application layer services.

UDP scan sends UDP messages prompting a UDP (in case of open port) or an ICMP response.

SCTP INIT scan utilizes SCTP INIT chunks prompting an INIT|ACK response in case of an open port, or ICMP error response otherwise.

The following scan options try to exploit/circumvent protocol peculiarities and surrounding infrastructure in order to retrieve more *precise* results for initially undetermined port status.

TCP NULL/FIN/Xmas scan tries to bypass the firewall by setting atypical or nonsensical TCP flag combinations within requests in order to examine the host's reaction. A FIN scan sends a FIN message to the targeted host, which may return an RST in case of a poorly configured non-stateful firewall. For NULL scans no flags are set, whereas for Xmas scan every flag is set.

TCP Maimon scan is similar to a FIN scan, except it sends a FIN|ACK "request". Host must return an RST with information about port status. Some BSD-derived systems drop this message in case of an open port.

Custom TCP scan allows setting arbitrary flags.

TCP ACK scan tries to map firewall rules by sending TCP ACK messages. It only determines whether a port is filtered (RST received?) or not.

TCP Window scan sends ACK requests similarly to ACK scan, however it also considers TCP window size, which enables the distinction between a closed and an open port. Positive window size within an RST response suggests an open port, zero size a closed one.

SCTP COOKIE ECHO scan detects whether a port is closed or not. It sends a SCTP COOKIE ECHO message. If the port is closed, an ABORT message is received. The host must drop the packet for an open port, thus making it indistinguishable from a filtered one.

TCP idle scan works through exploiting the so called zombie host, that utilizes incremental IP ID generation. First, the scanning entity contacts the zombie host to check its IP ID generation strategy. Then, the scanning entity masquerades as a zombie host by spoofing its IP address and sends a SYN request to the target. The target then responds (or not) to the zombie host with SYN|ACK or RST. Finally, the scanning entity once again contacts zombie for checking its IP ID. If it has increased by one, then the port is open, which means, that the zombie host has "surprisingly" received a SYN|ACK and answered with an RST incrementing its internal IP ID variable. Otherwise the targeted port is either filtered or closed. The intention behind the idle scan is to check the port status while remaining completely invisible to the targeted host.

IP protocol scan iterates over Internet protocol numbers, in order to find supported protocols. Should there be any encapsulated Transport layer response, its port will be included in the report.

FTP bounce scan exploits the FTP PORT call, which – where supported – allows the usage of a *remote* FTP server running in passive mode to check port status of the targeted host by sending files to its ports. As with the TCP idle, the idea behind bounce scan is not to disclose oneself to the host.

Nmap also has various parameters to adjust the accuracy–stealthiness–speed tradeoff. The `-T`*n* parameter allows to choose one of the six presets (numbered 0–5). Each preset includes predefined settings for round-trip timeout, packet delay and parallelism. For instance, a `-T0` or *paranoid* scan will wait 5 min (!) before sending each packet in order to evade intrusion detection/protection systems. By contrast, a `-T5` *insane* scan does not practically involve any artificial delay, sending packets to multiple hosts in parallel with a minimal RTT timeout, thus tolerating only sufficiently fast responses. `-T5` suggests usage on a very fast network, preferring speed over accuracy and stealthiness.[24]

3.2 ZMap

ZMap is being developed by Zakir Durumeric, Eric Wustrow, and J. Alex Halderman at the University of Michigan. It was initially released in August 2013,[12] and, since then, it has received considerable coverage by media and other academic researchers.[2][10][31] The reason for this is the claim to perform a full Internet-wide scan in under 45 minutes.[12]

Unlike the general-purpose Nmap, ZMap was custom-built with Internet-wide scans in mind. As a result, ZMap is far less customizable. Despite having functionality for group scans, developers state the *horizontal* scans as its main modus operandi. In order to achieve such short times, ZMap differs from Nmap by utilizing the following features:[12]

Probe randomization. As we already mentioned before, scan performance depends on network bandwidth. Randomizing the order in which probes are sent helps to avoid bandwidth saturation. Using a random order will drastically decrease the probability of simultaneously sending probes to hosts which belong to the same network, and, as a consequence, overloading this network's infrastructure will become far less likely. ZMap uses address space permutation via multiplicative cyclic group modulo *prime* p: $(\mathbb{Z}/p\mathbb{Z})^\times$ with $p = 2^{32} + 15$. This group reaches the entire IPv4 space except for 0.0.0.0. For each scan ZMap generates a new primitive root g and randomly chooses the initial IP address a_0. The next value a_{i+1} is then calculated with $a_{i+1} = (a_i \cdot g) \mod p$. ZMap also may utilize *sharding* in order to split the workload between n threads. In this case g^n is taken into calculation, resulting in $a_{n(i+1)+j} = a_{ni+j} \cdot g^n \mod p$ for the j-th shard $(0 \leq j < n)$.[1]

Asynchronous design. Unlike Nmap, ZMap *does not keep state* in order to match responses to requests. Instead, sending and receiving packets happens independently in separate threads, which in turn increases overall performance. Thus, the sending thread "forgets" about the connection immediately after it has been initiated. However, it is still possible for receiving threads to get to know, if the incoming response was intended for ZMap. In a similar manner to SYN cookies, ZMap calculates a UMAC using a scan-specific key over the destination IP address. The UMAC value is then written into available probe fields, e. g. source port and SEQ in case of TCP. Should there be a response to this probe, the receiving thread will be able to verify the request origin by comparing the destination port and decremented ACK field value with computed UMAC over the IP source field.[12][1]

No retransmission. ZMap tolerates packet loss for the sake of performance. ZMap sends a fixed number of probes to the target, yet a single packet is sent by default. Durumeric et al. concluded, that there are no significant losses with this setting.

Besides, ZMap is able to use the PF_RING ZC driver in order to bypass kernel routines and construct Ethernet frames on its own. This allows to exceed the 1 GbE bottleneck of the Linux kernel (assuming faster connection, e. g. 10 GbE).[1]

3.3 Masscan

Masscan is being developed by security researcher Robert David Graham, with its initial release dating back to October 2013. Graham went so far as to say masscan would perform an Internet scan within 3–6 minutes, assuming a 10 GbE connection.[15] Since then, there has been a silent rivalry between Graham and ZMap developers. Graham stated that ZMap's speedup over other port scanners is due to improvements in the Linux kernel.[16] In response, Durumeric et al. tried to experimentally disprove masscan's advance in their works.[1]

Masscan uses the same asynchronous model – splitting responsibilities for sending and receiving between threads – as scanrand, unicornscan, or ZMap. The main advantage of masscan lies in **probe randomization**. As with ZMap, the aim is to distribute IP addresses on-the-fly while iterating over the IPv4 address space. In his implementation Graham relinquished a certain degree of statistical randomness to save computation time for very high packet rates. At the rate of 10 million packets per second (Mpps), one has roughly 100 ns for packet processing. In order to beat this time masscan uses the *BlackRock* algorithm – a modified implementation of symmetric encryption algorithm DES with less rounds and modulo operations in place of binary ones allowing arbitrary ranges. Since DES is a Feistel cipher with substitution boxes, the ciphertext blocks appear as uniformly distributed pseudorandom bit strings.[15] Graham has also spoken about improving statistical distribution while retaining performance in the future.[14]

Masscan is based on the C10M paradigm proposed by Graham – handling 10 million connections at 10 Gbps / 10 Mpps with 10 µs latency, etc. Graham calls for removing the network routines from kernel in order to achieve this goal. Masscan may optionally utilize PF_RING ZC driver to bypass kernel and use its own TCP/IP stack instead.[13]

4. SCANNING ENTITIES

In the following we will introduce several scanning entities that perform Internet-wide scans. Since the persons behind some scan initiatives are actually unknown, we use the name of event for describing them.

CAIDA The Center for Applied Internet Data Analysis (CAIDA) is a research institute based in San Diego, California. CAIDA emerged in 1998 as a cooperation between the San Diego Supercomputer Center (SCDC) at UC San Diego as well as various commercial and governmental organizations. The main purpose of CAIDA is measurement, monitoring, and analysis of the Internet infrastructure. Though it is merely one of many research fields, CAIDA is performing extensive active and passive measurements. For instance, port scan detection is performed on the so called UCSD Network Telescope – a "globally routed /8 network" – using packet capture. For active port scanning CAIDA uses the task-based tool *scamper* being developed by Matthew Luckie. Scamper supports TCP, UDP and ICMP probing as well as pinging and traceroute. Scan results and visualizations are available at `http://www.caida.org/data/`.[4]

University of Michigan The developers of ZMap are maintaining the "Internet-Wide Scan Data Repository" publicly available at `https://scans.io`. The website is hosting the scan results done by ZMap team, but also by any third party willing to publish their research data.

Project Sonar is a community effort guided by Rapid7, a US company specializing on IT security. The project was initially defined by horizontal IPv4 Internet-wide scans on port 443 (HTTPS) using ZMap, and, furthermore, by collecting SSL/TLS certificates. However, Project Sonar was later expanded to UDP scans as well as gathering HTML index files from port 80 and DNS records. The goal of the project is to deliver a global view on the security of web services. The results are hosted by the aforementioned ZMap Team's scans.io repository.[27]

SIPscan was an Internet-wide group scan on UDP ports 5060, 5061, 5070 and TCP port 80. SIPscan was detected and observed by UCSD Network Telescope for twelve days in February 2011. The name originates from the Session Initiation Protocol (SIP) which operates on these UDP ports. SIP is mainly used for Internet telephony (voice and video calls). The protocol is famous for being vulnerable to a great variety of attacks. SIPscan was a globally distributed scan performed by a botnet spanning over 3 million IPv4 addresses. Several computers worldwide were co-opted into the botnet via Sality malware, however the initiator(s) of SIPscan remain(s) unknown.[8]

Conficker traffic is another prominent example of using port scanning for finding vulnerable hosts. Conficker is the name of a computer worm, which exploits the Server service vulnerability in Windows operating systems. First, the attacker initiates an SMB session on the TCP port 445 of the victim. Then the attacker plants malicious code into the request which forces the victim to download an executable file. After analyzing captured Conficker traffic, the ZMap Team has reported, that 445/TCP scans are rather short-range horizontal scans. Additionally, Conficker traffic is the dominant port scanning initiative among small scans (targeting $< 10\,\%$ of IPv4 space).[11]

Internet Census 2012 was another Internet-wide scan using a botnet – dubbed Carna Botnet – of roughly 420.000 embedded devices (mostly routers with either default, weak, or even no password). These devices were supplied with lightweight Nmap builds, each scanning a small host range. Internet Census 2012 is a distributed Internet-wide group scan on 150 most used ports. The initiator chose to remain anonymous, however an extensive description, evaluation, and visualization of results along with scan data is available at `http://internetcensus2012.bitbucket.org`.[17]

Shodan is a controversial search engine created 2009 by the "Internet cartographer" John Matherly for finding various devices connected to the Internet. Since CNNMoney released an article about a vast amount of insufficiently protected services publicly available on the Internet and disclosed by Shodan, spanning from baby cams to power station remote control systems, there have been major concerns about legal and ethical background of the service. Security expert Richard Bejtlich called Shodan an "intrusion as a service" (as a reference to various cloud service delivery models). For its continuous scans, Shodan utilizes its own software simply named "Shodan crawler", which does both port scanning and banner grabbing. Shodan also provides a scan result repository ScanHub, where users can upload their Nmap/masscan XML files for searching and visualization purposes, but also to make it available publicly in Shodan's search results. An interesting fact is that the Internet Census 2012 was only possible due to the variety of unprotected devices on the Internet, whereas Shodan drew media attention to the possible extent of such botnets.[30]

NSA/GCHQ HACIENDA is the name for a reconnaissance program led by the US National Security Agency (NSA) and the UK Government Communication Headquarters (GCHQ) – both governmental intelligence organizations of their respective countries. The program was classified top secret, however, in August 2014, presentation slides leaked into the public domain. HACIENDA aims at port scanning of country IPv4 ranges. Port scans were complete for 27 countries. According to slides, HACIENDA staff also takes orders from associates for scanning countries originally not on the list. HACIENDA uses Nmap as scanning tool with host randomization parameter.[20]

Open Resolver projects aim at finding poorly configured recursive DNS servers (scanning for port 53), which may be be misused for DNS amplification attacks – a form of DDoS attack performed via flooding the target with DNS responses. The intention of the project is to provide an overview over vulnerable DNS hosts, thus encouraging system administrators to properly configure their servers. Open Resolver projects carried out by different groups of individuals, the Shadowserver Foundation among others. Aggregated reports are available at `https://dnsscan.shadowserver.org/` and `http://openresolverproject.org/`. Raw data is only provided by request.

5. ENTITY CLASSIFICATION

In this section we will propose a categorization model for scanning entities. The main purpose of this model is to give a compact but extensive description of a scanning entity, so that both similarities and differences between individual entities become clearly visible. The following model is heavily based on examining the behavior and the background of a particular entity. We decided to classify entities by the following attributes:

Organization/institution says a lot about the scanning entity. In fact, further attributes heavily depend on the type of organization the scanning entity is representing. Clear attribution is only possible if the entity is publicly known. However, in most cases scanning entities choose to remain anonymous. In this case we use a generic term "individual". We propose the following categories:

1) *academic*
2) *governmental*
3) *commercial*
4) *individuals(s)*

Intention behind Internet-wide scans is the most important aspect when speaking about legality and ethics. There are basically two major reasons for performing a scan – finding vulnerabilities to exploit and doing research/auditing to provide a global overview. But despite that, we found out, that Shodan and HACIENDA clearly stand out from other entities. Shodan neither does research nor wants to exploit any vulnerabilities, whereas HACIENDA is a reconnaissance program, which is different from the common understanding of vulnerability exploits as a means to gaining personal profit. We also want to emphasize the presence of *military* operations among port scanning activities. Thus, we chose to add a third category:

1) *exploit*
2) *research / security audit*
3) *discovery/reconnaissance*

Spatial distribution of scan sources is another attribute which defines a scanning entity. As an example, Internet Census 2012 was performed from a worldwide botnet of constrained devices.

1) *single host / local cluster*
2) *wide-area/globally distributed*

Publication of results is important for researchers who either want to use the data for their own work, or to verify the work based on this data. As described in section 2.3 particularly metadata may give crucial knowledge about the targeted hosts. We distinguish three degrees of publication extent:

1) *undisclosed*
2) *aggregated report* (w/o raw data)
3) *complete*

Used software category tells, whether these tools are available to public:

1) *enterprise*
2) *publicly available* (incl. custom builds)

Timing describes if a scan initiative is still present or has taken place in the past:

1) *passed*
2) *ongoing*

Table 1 presents classification of the aforementioned scanning entities. Note, that for SIPscan and Conficker we only consider malicious intents. Security auditing must be examined separately. Additionally, some behavior patterns can be seen: An academic institutions is most likely doing research and hence will also publish scan results (University of Michigan being a good example). Attacks on vulnerable services (here: Conficker, SIPscan) include a distributed approach – botnets. By contrast, Internet Census 2012 bears all the hallmarks of a research undertaking, but it utilized a botnet which is more common for malicious activities. Almost every entity is using publicly available software with the single exception of Shodan. However, Shodan also processes XML reports from Nmap and masscan which were uploaded by ScanHub users.

6. LEGAL STATUS & ETHICS

As we already mentioned before, port scanning affects both scanning and scanned entities. Port scanning has been a controversial topic for more than two decades.[23] Some consider it a harmless examination, while others regard port scanning as an intrusion. Roughly summarized, there are two contrary points on this topic:[5]

	CAIDA	UMich	Sonar	SIPscan	Conficker	IC2012	Shodan	HACIENDA	Open Resolver
organization									
academic	■	■							
governmental	■							■	
commercial	■		■				■		
individual(s)			■	■	■	■			■
intention									
exploit				■	■			■	
discovery/recon							■	■	
research/audit	■	■	■			■			■
distribution									
single/local	■	■	■				■	■	
wide/global				■	■	■			■
publication									
undisclosed				■	■			■	
aggregated									■
complete	■	■	■			■	■		■
used software									
enterprise							■		
public	■	■	■	■	■	■	■	■	■
timing									
passed				■		■			
ongoing	■	■	■		■		■	■	■

Table 1: An attempt to classify Internet-wide scan initiatives and scanning entities

- *A port scan is the precursor to an attack.*
- *A port scan is an act of curiosity about services, which are offered to the public anyway.*

The ethical debate apparently looks like a matter of taste at the first sight, however, it is the key to understanding the legal situation concerning the port scanning in most countries. In the following we will briefly introduce the legal situation in a few chosen areas of jurisdiction.

One of the most famous legal cases involving port scanning is Moulton v. VC3. Network engineer Scott Moulton was appointed to maintaining a municipal emergency network in Georgia, US. Moulton was concerned about network security as he launched a port scan for security auditing a network he had been previously setting up. Eventually his scan reached the IP addresses of the consulting firm VC3. Since Moulton did not conceal his identity, VC3 informed the police, whereupon Moulton was arrested. He was sued for violation of the Computer Fraud and Abuse Act (CFAA) as well as the Georgia Computer Systems Protection Act. After months of litigation, the court ruled, Moulton did not violate CFAA, hence all charges against him were dropped.[23] Severe violation of CFAA may be punished with up to 10 years of imprisonment [18 U.S.C. § 405(c)(1)(A)].

Laws dealing with computer fraud exist all around the world, however explicit mentioning of port scanning is far less likely to find. In 2006 the UK Computer Misuse Act 1990 was amended with the Police and Justice Act 2006, that prohibited “supplying or obtaining articles for use in computer misuse offences”.[Computer Misuse Act 1990 (amended by Police and Justice Act 2006) F63A] For example, downloading the “network stress testing application” Low Orbit Ion Cannon (LOIC) used mainly for denial-of-service attacks (DoS) is a crime within the UK jurisdiction. The legal phrasing “computer misuse offences” is vague, thus any port scanning tool may be considered as such an “article”. By contrast, the § 202c of the German penal code clearly prohibits the use of *purpose-built intrusion* software.[§ 202c StGB (German penal code) Abs. 1] This, however, is also open to interpretation, since the use of dubious scan methods such as idle scan or FTP bounce may be considered an intrusion. Another point is afflicting damage to the targeted host(s). Aggressive scanning as well as some scan parameters may overload the network or even crash some hosts. This can be surely considered as an unintentional tort (prosecuted in most countries with a fine) or even a DoS attack (computer fraud laws apply).

Aside from legal implications, there are several ISPs that prohibit port scans. US-based cable company Comcast explicitly prohibits port scanning for its customers.[7] German ISP Deutsche Telekom does not allow establishing connection to a remote host as an end in itself.[9] Violating terms of use will most likely result in contract void, but rather rarely in further legal disputes.

The specific character of Internet-wide scanning implies, that a scanning party may break several laws in multiple countries. Depending on the severity of law violation and international criminality treaties, the home country may be obliged to extradite the criminal suspects. Scottish system administrator Gary McKinnon was accused of computer fraud by the US authorities after port scanning and exploiting vulnerabilities of several US military organizations and nearly faced extradition[28] until the order was withdrawn in 2012 by the UK Home Secretary.[19] Though, this was a clear case of intrusion, port scans are definitely portraying a suspicious activity for military organizations – whatever consequences that means.

7. CONCLUSION

In this paper we have introduced the state-of-the-art publicly available software tools for Internet-wide scans. The classic among port scanners Nmap, while highly tunable and accurate, cannot reach the speed of the newcomers ZMap and masscan – specialized software for running /0 scans.

In order to answer the question "Who is scanning the Internet?" we presented various organizations and individuals who are openly performing port scans on a large scale. We learned that they pursue different goals, with the most interested in security/exploits, but some of them also being strikingly different, namely Shodan and the HACIENDA program. In order to systematically organize scanning entities by their attributes we proposed a classification model. The model exhibits some interesting behavior patterns concerning the various attributes of individual entities.

Finally, we discussed some some legal and ethical aspects of Internet-wide scanning. We focused on striking abuse allegations due to port scanning with further elaborating on consequences on legality of performing it on a large scale. We concluded that the vagueness of legal acts may present a serious burden for those willing to perform such scans on their own.

8. REFERENCES

[1] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman. Zippier zmap: internet-wide scanning at 10 gbps. In *Proceedings of the 8th USENIX Workshop on Offensive Technologies*, 2014.

[2] J. Blagdon. Take a snapshot of the entire internet in just 44 minutes with ZMap scanner. *The Verge*, 2013.

[3] B. Burns, J. S. Granick, S. Manzuik, P. Guersch, D. Killion, N. Beauchesne, E. Moret, J. Sobrier, M. Lynn, E. Markham, C. Iezzoni, and P. Biondi. *Security Power Tools*. O'Reilly, 2007.

[4] CAIDA. Frequently asked questions about caida. `http://www.caida.org/home/about/faq.xml`, January 2015.

[5] Chaos Computer Club Cologne e. V. Warum wir nichts gegen portscans haben. `http://koeln.ccc.de/ablage/portscan-policy.xml`.

[6] R. Clayton. Brute force attacks on cryptographic keys. `http://www.cl.cam.ac.uk/~rnc1/brute.html`.

[7] Comcast. Acceptable use policy. `http://business.comcast.com/customer-notifications/acceptable-use-policy`, February 2013.

[8] A. Dainotti, A. King, F. Papale, A. Pescape, et al. Analysis of a/0 stealth scan from a botnet. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 1–14. ACM, 2012.

[9] Deutsche Telekom. Allgemeine geschäftsbedingungen. festnetz- und mobilfunkanschlüsse (privatkunden). `http://www.telekom.de/dlp/agb/pdf/40810.pdf`, December 2012.

[10] P. Ducklin. Welcome to Zmap, the "one hour turnaround" internet scanner. *Naked Security*, 2013.

[11] Z. Durumeric, M. Bailey, and J. A. Halderman. An internet-wide view of internet-wide scanning. In *USENIX Security Symposium*, 2014.

[12] Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast Internet-wide scanning and its security applications. In *USENIX Security*, pages 605–620, 2013.

[13] R. D. Graham. The C10M problem. `http://c10m.robertgraham.com/p/manifesto.html`, 2013.

[14] R. D. Graham. Masscan: designing my own crypto. `http://blog.erratasec.com/2013/12/masscan-designing-my-own-crypto.html`, December 2013.

[15] R. D. Graham. Masscan: Mass IP port scanner. `https://github.com/robertdavidgraham/masscan`, 2013.

[16] R. D. Graham. Masscan: the entire Internet in 3 minutes. `http://blog.erratasec.com/2013/09/masscan-entire-internet-in-3-minutes.html`, September 2013.

[17] IC2012. Internet Census 2012: Port scanning /0 using insecure embedded devices. `http://internetcensus2012.bitbucket.org/`, 2012.

[18] Internet Society. Measurements | world ipv6 launch. `http://www.worldipv6launch.org/measurements/`, July 2015.

[19] M. Kennedy. Gary McKinnon will face no charges in UK. `http://www.theguardian.com/world/2012/dec/14/gary-mckinnon-no-uk-charges`, December 2012.

[20] J. Kirsch, C. Grothoff, M. Ermert, J. Appelbaum, L. Poitras, and H. Moltke. NSA/GCHQ: The HACIENDA Program for Internet Colonization. `http://heise.de/-2292681`, August 2014.

[21] C. Leckie and R. Kotagiri. A probabilistic approach to detecting network scans. In *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*, pages 359–372. IEEE, 2002.

[22] C. B. Lee, C. Roedel, and E. Silenok. Detection and characterization of port scan attacks. *Univeristy of California, Department of Computer Science and Engineering*, 2003.

[23] G. Lyon. *Legal Issues*, chapter 1. Nmap Project `http://nmap.org`, 2009.

[24] G. Lyon. *Nmap Reference Guide*, chapter 15. Nmap Project `http://nmap.org`, 2009.

[25] V. Paxson. Strategies for sound Internet measurement. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 263–271. ACM, 2004.

[26] E. Raftopoulos, E. Glatz, X. Dimitropoulos, and A. Dainotti. How dangerous is internet scanning? In *Traffic Monitoring and Analysis*, pages 158–172. Springer, 2015.

[27] Rapid7. Project sonar by rapid7. `https://sonar.labs.rapid7.com/`, 2013.

[28] G. Roberts. Gary McKinnon: Inside the head of a super hacker. `http://www.independent.co.uk/news/science/gary-mckinnon-inside-the-head-of-a-super-hacker-407656.html`, July 2006.

[29] Sciengines. Copacobana: A Codebreaker for DES and other Ciphers. `http://www.sciengines.com/copacobana/index.html`.

[30] Shodan. Shodan: The search engine for Internet of Things. `https://www.shodan.io/`, 2015.

[31] I. Thomson. New tool lets single server map entire internet in 45 minutes. *The Register*, 2013.

[32] Y. G. Zeng, D. Coffey, and J. Viega. How vulnerable are unprotected machines on the internet? In *Passive and Active Measurement*, pages 224–234. Springer, 2014.