How Vulnerable are Unprotected Machines on the Internet?

Yuanyuan Grace Zeng[1], David Coffey[2], and John Viega[1]

[1] SilverSky yzeng@silversky.com, jviega@silversky.com
[2] McAfee,Inc. david_coffey@mcafee.com

Abstract. How vulnerable are unprotected machines on the Internet? Utilizing Amazon's Elastic Compute Cloud (EC2) service and our own VMware ESXi server, we launched and monitored 18 Windows machines (Windows 2008, XP and 7) without anti-virus or firewall protection at two distinct locations on the Internet—in the cloud and on-premise. Some machines ran a wide-open configuration with all ports open and services emulated, while others had a out-of-the-box configuration with default ports and services. After launching, all machines received port scans within minutes and vulnerability probes within a couple of hours. Although all machines with wide-open configurations attracted exploitations within a day, machines with out-of-the-box configurations observed very few vulnerability exploitations regardless of their locations. From our months-long experiment we found that: a) attackers are constantly searching for victims; b) the more opening ports/listening services a machine has, the more risks it is exposed to; c) brute-force logins are the most common type of attack; d) exploitations targeting vulnerabilities of software or operating systems are not widely observed.

1 Introduction

The Internet is a playground for opportunistic attackers. Thousands of threats are circulating around the Internet. Most computers today are protected by firewalls, IDS/IPS and anti-virus (AV) tools. But what happens in the worst-case scenario when they do not have any protection? Previous experiments on "Time-to-Live-on-the-Network" [5] and "Survival Time" [11] of Windows machines were conducted quite a few years ago with test machines running old Windows operating systems. The Internet Storm Center of SANS made the "Four-Minute Windows Survival Time" [10] claim in 2008 and was especially criticized for using a Windows XP RTM or SP1 version in the test.

Since the time of these initial time-to-live studies, the Internet threat environment has become deadlier. Meanwhile, the Windows operating systems have become more secure. But in the past five years, we failed to see any study on attacks towards unprotected machines running current operating systems. To close this gap, we wanted to investigate how well an unprotected machine with a current operating system does in today's threat environment. Left to its own devices, how soon will it be probed and attacked? And what is the most prevalent attack targeting the unprotected machine? We are interested in testing unprotected machines at two places on the Internet: one is in the cloud and the other is on-premise connecting directly to a DSL line. We would like to study the in-cloud scenario because enterprises are increasingly turning to the cloud for

various business purposes. Also, since Windows operating systems account for more than 80% market share [3], we would like to focus our study on the most widely-used Windows operating systems. To the best of our knowledge, this is the first experiment carried out on Windows 2008 and Windows 7 machines. Unlike previous experiments that only captured the elapsed time for a machine to get infected, our experiment kept track of different stages of a malware infection process. We measured the time elapsed starting with the initial deployment of the test machine to the first occurrence of all following events: port scan, vulnerability probe, and exploitation. Based on detailed traffic and event logs captured, we were able to conduct a thorough analysis on the scan/probe/exploitation activities.

2 Related Work

Besides the aforementioned empirical time-to-live studies on Windows machines, there are other areas of research related to our work. One such area is vulnerability assessment. Ten *et al.* [13] proposed a framework to quantify and evaluate the vulnerabilities of SCADA systems at multiple levels. Hartung *et al.* [6] demonstrated the ease of compromising a sensor node and tampering its data, and suggested a few countermeasures to improve a sensor's security posture. McQueen *et al.* [7] created a time-to-compromise model for a system component that is visible to an attacker, taking into account known and visible vulnerabilities, and attacker skill level. Another relevant topic is the analysis on the Internet-wide malware propagation. Moore *et al.* [8] conducted a case study on the infamous Code-Red worm at the global level, detailing the spread of this worm and the properties of the infected machines. Shannon *et al.* [12] monitored the outbreak of Witty worm through a network telescope and reported findings such as the scanning rate, the infection duration as well as the number of victims over a period of time. Moore *et al.* [9] studied the use of public search engines to locate vulnerable servers and found that as an alternative to vulnerability scanning this approach was widely used in compromising web servers to host malware and phishing sites.

3 Experiment Design

3.1 Scope of the Experiment

Usually, a machine gets infected through either of the two ways: user-involved infection or vulnerability exploitation. A user-involved infection requires a user to take certain actions such as clicking a link or downloading and executing a file. An infection via vulnerability exploitation normally gets its way into the machine silently without a user's awareness. Our experiment only considers the vulnerability exploitation scenario with no user in the loop. Every Windows machine in our experiment meets the following requirements:

- Each machine is connected to the Internet with a unique public IP address.
- All incoming traffic (TCP, UDP and ICMP) is allowed by a network-based firewall.
- The in-host Windows firewall is disabled and no anti-virus (AV) is installed.

– Wireshark captures all network traffic; Regshot [4] monitors Registry changes and Windows event logs keep track of system-wide activities such as logins/logouts and application status changes. Those logs together are used to decompose probes and attacks.

3.2 Experiment Set-Up

Our experiment spanned two periods of time: February to April and August to October of 2012. We set up and collected data from 18 machines in total at two locations—the Amazon's Elastic Compute Cloud (EC2) and a VMware ESXi server on-premise.

In-Cloud Experiment We ran 15 machines in Amazon's EC2 environment with two configuration profiles: "wide-open" and "out-of-the-box". In the wide-open scenario, a machine opens all ports and emulates all possible services. This way the machine can attract as many malicious attempts as possible. In the out-of-the-box scenario, a machine runs only with default open ports and services. This scenario gives us a baseline of how many malicious attempts an unprotected machine might encounter.

Windows is by far the most popular operating system on the Internet. Its server versions are generally exposed to more risks than home/professional versions. Our tests were carried out on Windows Server 2008 R1 SP2 and R2 SP1. As mentioned earlier, we disabled all firewall and anti-virus programs and configured the security policies so that Amazon allowed all incoming connections to those machines. To create the wide-open scenario, we installed a low-interactive honeypot named HoneyBot [1] and changed several services to avoid interference. After the configuration was complete, we took a snapshot of the instance and created an AMI (Amazon Machine Image) for later use. We launched ten instances on EC2 using the same AMI and made sure that they were hosted in different geographical zones and were allocated different IP addresses. For the out-of-the-box scenario, we made a clean install of Windows Server 2008 and did not install any programs other than Wireshark and Regshot. By default, common ports such as 135 (RPC), 139 (NetBIOS), 445 (SMB) and 3389 (RDP) were open. We ran five such instances on EC2.

On-Premise Experiment To create a testbed, we installed a VMware ESXi 5.0 server and connected it to a DSL line at our office location in North Carolina. This time we wanted to test out non-server Windows OS versions. Since OS platform statistics [3] showed that Windows 7 and Windows XP accounted for a majority of Windows operating systems being used (55% and 25% in August 2012), we created three virtual machines on the ESXi server: one running Windows 7 Professional SP1 and two running Windows XP Professional SP2. Their default open ports included 135 (RPC), 139 (NetBIOS), 445 (SMB) and 3389 (RDP). We later opened port 21 (FTP), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 1433/1434 (MSSQL) on those Windows XP machines. Each virtual machine was assigned a unique public IP and we ran port scans to confirm that machines were indeed reachable from the Internet. Other configurations were the same as the in-cloud machines.

4 Experiment Results

4.1 In-Cloud Experiment

Scan, Probe, and Exploitation Times of Occurrence Malware infections follow a predictable pattern. Using a port scan, an attacker tests whether a port on a target machine is open. If so, a vulnerability probe gathers more information about a listening service, such as the version of the service to identify specific vulnerabilities; and an exploitation delivers malicious payloads to finally compromise the machine. In the wide-open scenario, after launching, on average it took about 23.4 minutes to see the first port scan, and 56.4 minutes to see the first vulnerability probe (the exact number for each server shown in Figure 1). Probes hit well-known ports such as 22 (SSH), 23 (Telnet), 25 (SMTP), 80 (HTTP), 445 (SMB), 1080 (SOCKS Proxy), 1433 (Microsoft SQL Server) and 3389 (RDP). Looking at each server (honeypot) individually, we found that honeypots 1, 7, 8 and 9, which were hosted in the same zone on EC2, waited longer to see the first port scans and probes. We surmised that the IP space of that zone was new and not yet explored by attackers. With respect to exploitation time windows, we observed that almost all first exploitation attempts came in within 24 hours, with the average time being 18.6 hours (Figure 2). We captured exploitation attempts on port 445 (SMB), 1434 (Microsoft SQL Monitor), 2967 (Symantec AV) and 12147 (Symantec Alert Management System 2). Almost all exploitations during our months-long experiment were known threats. This is expected because the HoneyBot program was able to emulate many known vulnerabilities to attract attacks. Interestingly, exploits targeting five to even ten years old vulnerabilities were still floating around. For example, the attack at port 1434 was the Slammer worm dating back to 2003, and the stack overflow vulnerability at port 2967 was disclosed in 2006.

In the out-of-the-box scenario, it took an average of 13 minutes for the first port scan to arrive (Figure 3). Port scans hit ports such as 8080 (HTTP) and 1433 (Microsoft SQL Server). The first vulnerability probe arrived within 3 hours on average (Figure 3); all probes were login attempts to the Samba share (445) or via RDP (3389). We monitored the servers for a few weeks, but failed to see any exploitation attempts mainly due to the limited number of open ports (services).

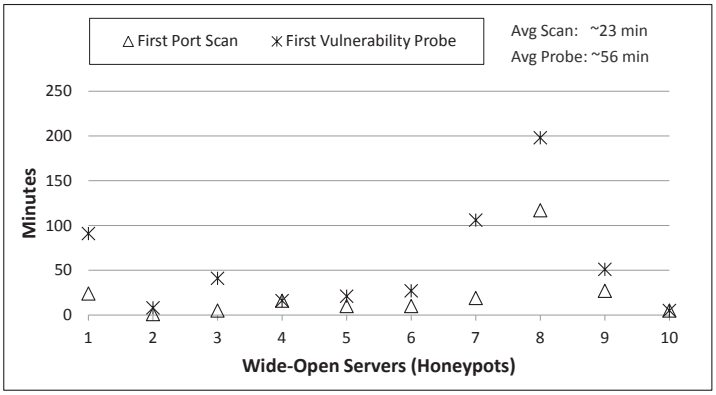

Fig. 1. Scan and Probe Times of Occurrence on Wide-Open Servers (in minutes)

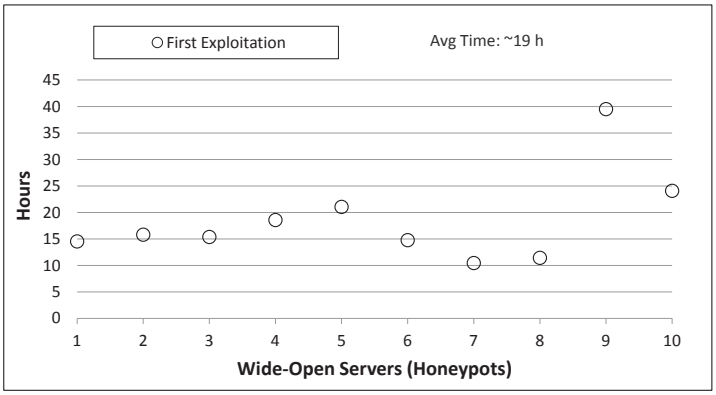

Fig. 2. Exploitation Attempt Times of Occurrence on Wide-Open Servers (in hours)

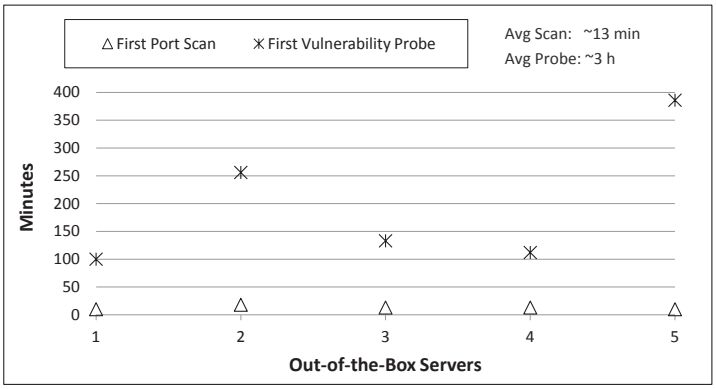

Fig. 3. Scan and Probe Times of Occurrence on Out-of-the-Box Servers (in minutes)

Top targeted ports In the wide-open scenario, all ports were open on each test machine. We analyzed the traffic to see which ports were targeted most often. Table 1 plots the top 10 ports ordered by the percentage of total traffic each port accounts for. As shown, 1080 (SOCKS) was the most targeted port. The SOCKS protocol is used to tunnel traffic through firewalls from inside to outside, but it is often misconfigured. Attackers take advantage of misconfigured SOCKS services to tunnel their attack traffic inward and mask the origin of their traffic—that's why this port attracted so many hits. Port 1433, the Microsoft SQL Server listener, also received much attack traffic. Port 25 (SMTP) was also popular. Spammers who look for open relays frequently probe this port. Many of the other top ports were related to the HTTP service, such as 80, 8000, 8080 and 8888. In the out-of-the-box scenario, we can see that (also in Table 1) more than 60% of traffic went to port 445 and 3389 which were open by default. Other common ports such as 1433, 80, 4899 and 1080, though not open, also received numerous scans.

Login Attempts In our experiment, we observed a huge number of login attempts. Almost all of them were failures according to Windows security event logs. Every test machine, on average, received over 1,000 login requests daily either through port 445/139

Table 1. Top 10 Targeted Ports

Wide-Open		Out-of-the-Box	
Port	% of Conn	Port	% of Conn
1080 SOCKS	15.50%	445 SMB	32.26%
445 SMB	10.94%	3389 RDP	28.85%
1433 MSSQL	8.03%	38856	6.07%
3389 RDP	6.29%	139 NetBIOS	2.81%
80 HTTP	6.01%	1433 MSSQL	2.48%
110 POP3	3.18%	22292	1.73%
22 SSH	2.93%	80 HTTP	1.58%
25 SMTP	2.91%	4899 Radmin	1.13%
139 NetBIOS	2.83%	27977	0.93%
8000 HTTP	1.76%	1080 SOCKS	0.90%

(SMB/NetBIOS) or port 3389 (RDP). SMB (Server Message Block) is an application layer protocol that is mainly used for file sharing on Windows systems. It can run directly over TCP port 445 or run in the session layer via port 139 over TCP. RDP (Remote Desktop Protocol) provides remote desktop connections for Windows. We looked at failed login attempts at port 3389 on our test machines. As it turned out, multiple offending IPs tested out the same dictionary of usernames. Table 2 demonstrates this set of usernames. In particular, the username *administrator*—the default administrative account name—was brute-forced the most. Examining the SMB login attempts,

Table 2. Brute-Forced Usernames

Usernames			
1	administrator	root	test2
123	aspnet	server	test3
a	backup	sql	user
actuser	console	support	user1
adm	david	support_388945a0	user2
admin	guest	sys	user3
admin1	john	test	user4
admin2	owner	test1	user5

we observed that attackers tried several administrator name variations such as *admin*, *administrator* and *db2admin*. All of those attempts failed except for a few anonymous (guest) logons. Anonymous logins do not require a username or password to connect to the SMB server. This is an optional feature of SMB and should generally be disabled. Anonymous logins may pose a security risk to the system because a remote attacker could launch exploits to gain user privileges or even control of the affected system.

Exploitations In our experiment, we found that most exploitations attempted on our wide-open machines were not new attacks. There was one interesting attack we would like to highlight—an attack on port 12147 where Symantec's Alert Management System

2 (AMS2) service listens. AMS2 is a component of multiple Symantec products including Symantec AntiVirus Corporate Edition and Symantec Endpoint Protection. AMS2 has multiple known vulnerabilities. For example, in 2009 a remote-code-execution vulnerability of AMS2 allowed attackers to execute arbitrary commands by sending a crafted packet. Our honeypot captured one such packet—the attacker attempted to get a remote shell to create a VBScript in the target machine. We extracted and reorganized the exploit packet payload and found that the main purpose of the script was to download an executable named *winnew.exe* from the attacker, save it as *installer.exe* to the *C:* drive, and then run it. With the remote command shell, the attacker was able to do whatever he wanted to the target machine. Our honeypots also captured similar exploits targeting the same vulnerability, but with different payloads.

Table 3. Summary Statistics

Machine	Total Time	1st Port Scan	1st Probe	# of Compromises	# of Connections Daily	# of Offending IPs Daily
Win XP Pro SP2	14 days	50m	1h51m	1	453	69
Win XP Pro SP2	7 days	6m	1h37m	1	2372	54
Win 7 Pro SP1	29 days	3m	2h41m	0	618	45

4.2 On-Premise Experiment

Scan, Probe, and Exploitation Times of Occurrence In the on-premise experiment, Table 3 shows the summary statistics of the three virtual machines (Two Windows XP and one Windows 7) on an ESXi server. They were connected to a DSL line in our office location. They all received port scans within an hour, probes within a couple of hours, though only the two XP machines were eventually compromised by attackers. The average numbers of inbound connections on daily basis were different from one machine to another. Apparently, some offending IPs behaved more aggressively than others, which we will show later.

Top Targeted Ports As far as top targeted ports are concerned (Table 4), Windows XP and Windows 7 machines shared a similar set of targeted ports such as 1433, 3389, 445 and 139. The MSSQL port 1433 was disproportionally targeted due to the Microsoft SQL server installed on the XP machines. Note that many ports in the top list were never open in our experiment—attackers made constant requests to them simply because most likely services running on those ports had vulnerabilities. Thus, it is important for network administrators and IT security staff to secure those services at first.

Top Offending IPs Table 5 lists the top 5 offending IPs along with targeted ports per IP and the number of connections initiated. The observation is that, as opposed to scanning all/multiple ports on a machine, the attacker normally focused on one particular port (service). For example, the top one offending IP to XP machines initiated thousands of connections only to the MSSQL port 1433, whereas the top offending IP to the Windows 7 machine persistently reached out to SMB port 445/139. Apparently, brute-forced login attempts accounted for a majority of the incoming connections.

Table 4. Top 10 Targeted Ports

Windows XP		Windows 7	
Port	% of Conn	Port	% of Conn
1433 MSSQL	36.58%	139 NetBIOS	53.56%
3389 RDP	8.88%	445 SMB	23.10%
445 SMB	1.62%	3389 RDP	13.87%
5900 VNC	1.53%	1433 MSSQL	1.58%
139 NetBIOS	0.59%	5900 VNC	1.55%
25 SMTP	0.40%	22 SSH	0.76%
22 SSH	0.31%	23 Telnet	0.48%
4899 Radmin	0.22%	4899 Radmin	0.45%
110 POP	0.22%	8080 HTTP	0.45%
80 HTTP	0.19%	51595 UDP	0.41%

Table 5. Top 10 Offending IPs with Targeted Ports

Windows XP		Windows 7	
IP/Port	# of Conn	IP/Port	# of Conn
64.31.*.*	**6988**	**80.90.*.***	**5663**
1433	6988	*139*	2790
218.65.*.*	**1987**	*445*	2873
3389	1987	**184.154.*.***	**4628**
199.36.*.*	**1360**	*137*	2
1433	1360	*139*	4626
117.41.*.*	**1050**	**122.199.*.***	**1463**
1433	1050	*3389*	1463
159.226.*.*	**912**	**205.210.*.***	**682**
1433	912	*135*	1
		139	454
		445	227
		200.91.*.*	**623**
		135	1
		139	310
		445	312

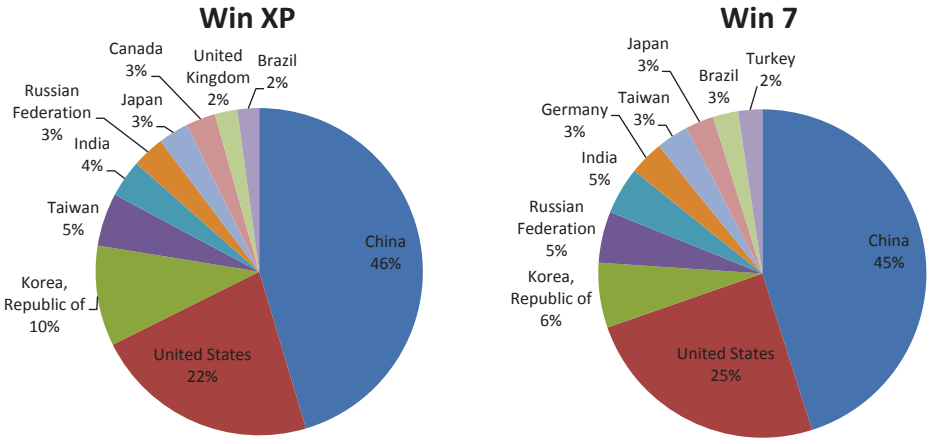

Fig. 4. Top 10 Countries of Attacks

Top Countries We used MaxMind's GeoIP database [2] to map source offending IPs to geographic locations. As shown in Figure 4, the top three countries remained the same for XP and 7 machines. The top one country is China, accounting for over one third of total malicious traffic. United States is at the second place and followed by Korea. We need to point out that the location of an offending IP does not necessarily reflect where the attacker is because an attacker can remotely control compromised machines all over the world.

Compromises As mentioned earlier, the Windows 7 machine stayed strong throughout the experiment, whereas the two Windows XP machines fell victim and were eventually under attackers' control. How did the compromises take place? Long story short: both were due to weak passwords as opposed to OS/software vulnerability exploitations. We will walk through them one by one.
Compromise I: The compromise of one XP machine was attributed to a Microsoft SQL Server brute-force attack. The intruder successfully broke the 'sa' account password ("password1") within 9 hours of service startup, and then enabled the *xpcmd_shell*, an extended stored procedure, to issue commands directly to the Windows command shell. With this privileged access, the machine was in the intruder's hand. The victim machine subsequently started FTP sessions with its command server and downloaded and executed multiple Trojan payloads. Instructed by the command server, the machine made numerous connection attempts to an online gaming site.
Compromise II: The compromise of another XP machine also resulted from a weak password ("tryout") of the Administrator login account. The intruder launched thousands of RDP sessions and finally made a right guess. It was about two days between our machine going online and being compromised. From the pcap traces, we could tell that the original intruder did not hold the machine for his own use. It seems that the compromised machine was given (or even sold) to someone else. Though the break-in method is a standard one, how the victim machine was used is noteworthy. There was no system change or file modification on the machine. We caught the wrongdoer at the scene the moment he was fabricating his eHarmony profile on the compromised machine. His IP was from Nigeria and there was a picture of an Italian actor on the desktop. Looking at the browsing history we found that this person had visited quite a few online dating sites to create new profiles and browse other peoples' pages—he logged on to this machine solely for this purpose. Given all the information, very likely, this is the starting point of an online dating scam. Why did he use someone else's machine to do so? Normally, web sites can track users by IP addresses and people conducting malicious activities are afraid of getting caught if using their own computers.

5 Conclusion and Future Work

In this paper, we presented our experiment on monitoring 18 unprotected Windows machines at two Internet locations: in the cloud and on-premise. Our key findings are:

– *Every machine on the Internet is scanned within minutes after connecting.* It does not matter whether a machine connecting to the Internet opens ports or not—any machine will be scanned within several minutes. This is not surprising because attackers don't know whether a port is open unless they scan it.

– *More open ports means more vulnerability probes.* The elapsed time between the machine startup and the arrival of vulnerability probes depends on the specific services that are running. The more listening services a machine has, the sooner it will be probed, and the more risks it will be exposed to.
– *More vulnerabilities means more exploitation attempts.* It is rare that attackers send exploitations blindly without first knowing that their targets are vulnerable. On the other hand, if unprotected machines have holes, chances are good that attackers will find them and attempt to exploit them. How long it takes depends on the vulnerabilities a machine has.
– *Brute-force logins are the most common type of attack.* We observed that brute-force login attempts were much more frequent than vulnerability probes or exploitations. On each machine, we captured dictionary attacks at port 445 (SMB) and 3389 (RDP), attempting thousands of username/password combinations. Most attempts targeted accounts with administrator privileges. Weak or default passwords can be easily broken and provide the best entry point.
– *Vulnerability exploitations without users' interaction are possible but not widely observed.* Even though every wide-open machine (all ports open and services emulated) received at least one vulnerability exploitation within hours, we saw very few exploitations on out-of-the-box machines. Generally speaking, exploitations coming directly from the Internet and targeting vulnerabilities of operating systems or applications are less prevalent nowadays—most exploitations are delivered at the client-side and require users' involvement such as opening a file or clicking a link.

As future work, we plan to broaden the scope of the experiment. We would like to 1) increase the number of test machines; 2) add other operating systems such as OS X and Linux; 3) deploy machines at more locations such as home and campus networks. We expect to run this experiment on an ongoing basis and regularly report our findings.

References

1. Honeybot. http://www.atomicsoftwaresolutions.com/honeybot.php
2. Maxmind geoip database. http://www.maxmind.com/en/geolocation_landing
3. Os platform statistics. http://www.w3schools.com/browsers/browsers_os.asp
4. Regshot. http://sourceforge.net/projects/regshot/
5. Avantgarde: Time to live on the network. Tech. rep. (2004)
6. Hartung, C., Balasalle, J., Han, R.: Node compromise in sensor networks: The need for secure systems. Department of Computer Science University of Colorado at Boulder (2005)
7. McQueen, M.A., Boyer, W.F., Flynn, M.A., Beitel, G.A.: Time-to-compromise model for cyber risk reduction estimation. Quality of Protection pp. 49–64 (2006)
8. Moore, D., Shannon, C.: Code-red: a case study on the spread and victims of an internet worm. In: 2nd ACM SIGCOMM Workshop on Internet measurment. pp. 273–284 (2002)
9. Moore, T., Clayton, R.: Evil searching: Compromise and recompromise of internet hosts for phishing. Financial Cryptography and Data Security pp. 256–272 (2009)
10. SANS: Four-minute windows survival time. http://isc.sans.edu/diary.html?storyid=4721
11. SANS: Survival time. http://isc.sans.edu/survivaltime.html
12. Shannon, C., Moore, D.: The spread of the witty worm. IEEE Security & Privacy 2(4), 46–50 (2004)
13. Ten, C.W., Liu, C.C., Manimaran, G.: Vulnerability assessment of cybersecurity for scada systems. Power Systems, IEEE Transactions on 23(4), 1836–1846 (2008)