Department of Computer Science

Seminar Sicherheitsmodellierung

ZMap

The Internet Scanner

Gehalten im:

Sommersemester 2014

Von:
Birte Wagner
Fachhochschule Wedel
Studiengang: Master Informatik
Matrikel-Nummer: 9952
E-mail: inf9952@fh-wedel.de

Dozent:
Prof. Dr. Gerd Beuster
Fachhochschule Wedel
Feldstraße 143
22880 Wedel
E-mail: gb@fh-wedel.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

ZMap ist ein Portscanner für internetweite Netzwerkstudien. Mit ZMap ist es möglich mit einer entsprechend schnellen Internetverbindung alle IPv4-Adressen des Internets in ca 45 Minuten zu scannen. Entwickelt wurde dieses Tool vor allem, um Sicherheitslücken zu entdecken und die Umsetzung von Fixes zu verfolgen.

1.1 Portscan

Ein Portscan eines Netzwerkes dient der Sicherheitsüberprüfung, aber auch dem Ausspähen von eigenen und fremden Rechnern und Netzwerken. Bei einem Portscan werden die Rechner über die IP-Adresse auf aktive Dienste (z.B. HTTP, TELNET, FTP) untersucht. Jeder dieser Dienste hat eine eigene Portnummer, mit der versucht wird sich zu verbinden. Grundsätzlich gibt es zwei Arten von Scans. Connect Scans versuchen einen Verbindungsaufbau zum Beispiel über TCP zu einem Dienst. Stealth Scans schicken ungültige Pakete an Dienste und versuchen aus der Antwort Informationen herauszulesen.

1.2 TCP SYN Scan

Der TCP SYN Scan ist ein halboffener Scan. Dies bedeutet, dass der Aufbau der TCP-Verbindung unterbrochen wird. Eine TCP-Connection wird über ein Three-Way-Handshake-Verfahren hergestellt.

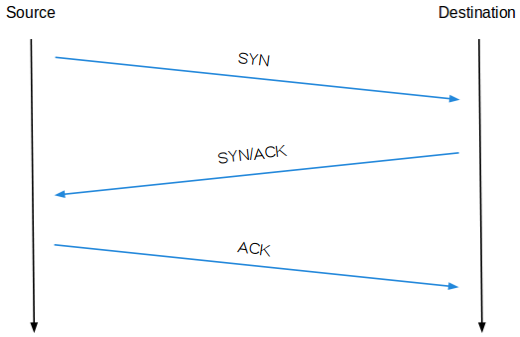

Abbildung 1.1: Erfolgreicher Three-Way-Handshake

In Abbildung 1.1 zu sehen ist, wird zunächst ein SYN-Paket an den Zielport des gewünschten Rechners geschickt, um zu erfahren, ob der Port offen ist. Ist der Port erreichbar, wird ein SYN/ACK-Paket zurückgeschickt. Der anfragende Rechner antwortet hierauf mit einem ACK-Paket, um den Verbindungsaufbau abzuschließen.

Ist der Zielport geschlossen, kommt statt eines SYN/ACK-Paketes ein RST-Paket zurück, zu sehen in Abbildung 1.2.

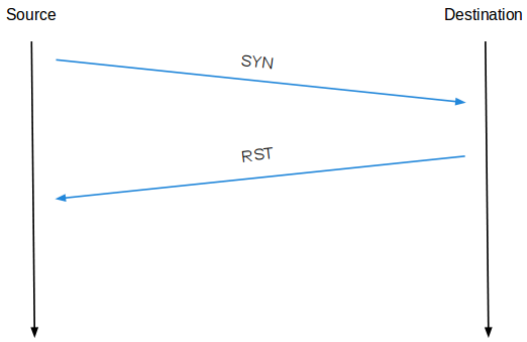

Abbildung 1.2: Kein Verbindungsaufbau durch einen geschlossenen Port

Bei einem TCP SYN Scan wird im letzten Paket anstatt dem ACK-Flag das RST-Flag gesetzt.

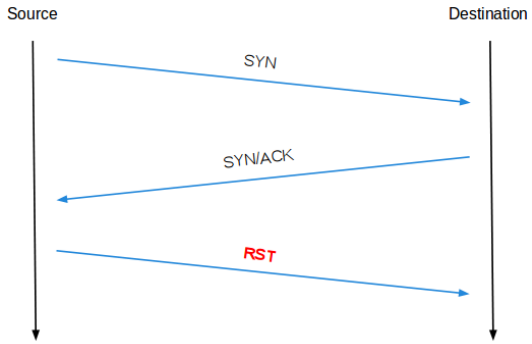

Abbildung 1.3: Handshake bei einem TCP SYN Scan

Hierdurch wird der Verbindungsaufbau abgebrochen. Von den meisten Rechnern werden TCP SYN Scans nicht mitgeloggt. Diese Art des Scannens reduziert die Anzahl der Pakete, die ausgetauscht werden müssen. Wenn der Host nicht erreichbar ist oder nicht antwortet, was meistens der Fall ist, wird nur das SYN-Paket vom Scanner gesendet. Bei einem geschlossenem Port werden zwei Pakete verschickt: Das SYN-Paket vom Scanner und ein RST-Paket als Antwort. Lediglich wenn der Port offen ist, werden drei Pakete verschickt: Die SYN-Anfrage vom Scanner, die Antwort ACK/SYN und der Abbruch durch ein RST-Paket vom Scanner.

2 Nmap

Nmap[1] ist der bekannteste und am häufigsten genutzte Portscanner. Nmap steht hier für Network Mapper. Nmap wurde 1997 entwickelt und seitdem immer wieder erweitert. 2001 gewann es den „Linux Journal's Editor's Choice Award"[2] als „Best Security Tool". Bei einem Besuch von George W. Bush 2006 bei der NSA war im Hintergrund ein Statusbildschirm mit Nmap zu sehen.

2.1 Verwendung

Nmap wird vor allem zur Netzwerkanalyse und Sicherheitsüberprüfung genutzt. Es ist sehr vielfältig und kann unter anderem anzeigen, welche Hosts im Netzwerk verfügbar sind, welche Dienste (Anwendungsname und -version) die Hosts bieten, welches Betriebssystem und Version auf dem Host laufen, welche Art von Paketfiltern/-firewalls verwendet werden und viele mehr. Verwendung findet Nmap vor allem in der Netzwerkadministration zum Beispiel für die Netzwerkinventarisierung und für die Verwaltung von Ablaufplänen für Dienstaktualisierungen. Auch zur Überwachung von Betriebszeiten von Hosts oder Diensten kann es, ähnlich wie Nagios, genutzt werden.

[1] Nmap Security Scanner [nmap]
[2] Linux Journal 2001 [LinuxJournal]

2.2 Funktionsweise

Die Übertragungsrate von Nmap ist soweit limitiert, dass die zu scannenden Netzwerke nicht überlastet werden. Hieraus resultieren die vergleichsweise schlechten Performancewerte im Vergleich zu ZMap. Hinzu kommt, dass Nmap mittrackt, welche Hosts bereits gescannt wurden bzw. von welchen Hosts noch eine Antwort aussteht. Durch diese Art von Tracking kann Nmap allerdings das Schließen von Connections durch Timeouts herausfiltern und hier einen weiteren Versuch des Verbindungsaufbaus starten. Nmap unterstützt unterschiedliche Scan-Techniken, die über Argumente ausgewählt werden können. Es kann volle TCP-Verbindungen aufbauen (-sT), einen TCP-SYN-Scan ausführen (-sS), UDP-Ports scannen (-sU), die Erreichbarkeit über Ping prüfen (-sP) und vieles mehr. Nmap kann auch absichtlich manipulierte oder falsche TCP-Pakete verschicken, um anhand der Reaktion zu prüfen, ob der Port offen oder von einer Firewall geschützt ist. Eine einfache Anfrage über die Konsole, die das lokale Subnetz scannen und jeweils 1000 Ports überprüfen soll, könnte folgendermaßen aussehen: nmap 192.168.1.0-255
Die Ausgabe auf der Konsole sah folgendermaßen aus:

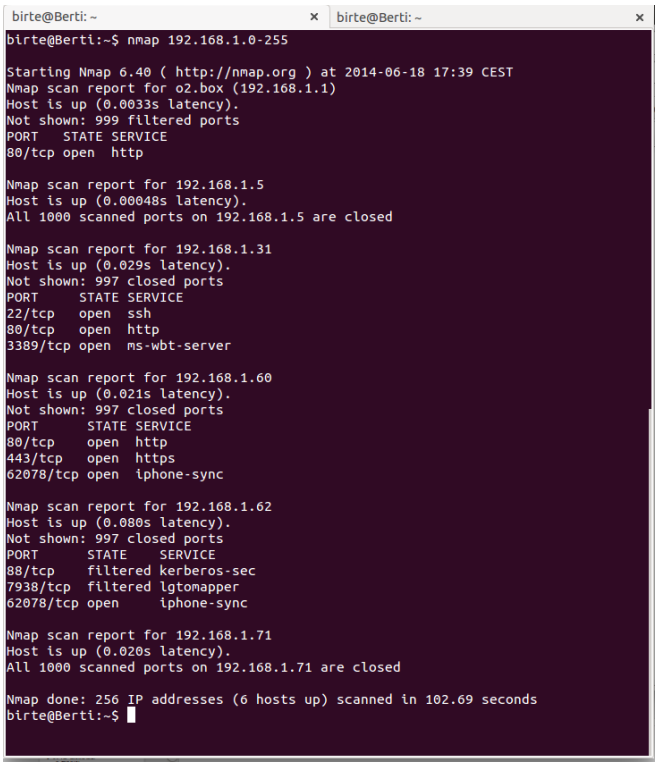

Abbildung 2.1: Konsolenausgabe eines Scans des lokalen Netzwerks mit Nmap

In der Abbildung 2.1 ist zu sehen, dass 6 Hosts, ein Router und 5 andere Devices gefunden wurden. Für jedes Device wurden aufgelistet unter welcher IP-Adresse es erreichbar war, welche Ports offen waren und welchen Status diese Ports hatten.

2.3 Zenmap

Zenmap[3] ist eine GUI für Nmap. Die Ergebnisse eines Scans werden hier grafisch aufgearbeitet und können auch gespeichert und wieder geladen werden. Hier ein Screenshot von der Ausgabe eines Scans mit dem gleichen Aufruf wie in dem vorherigen Beispiel:

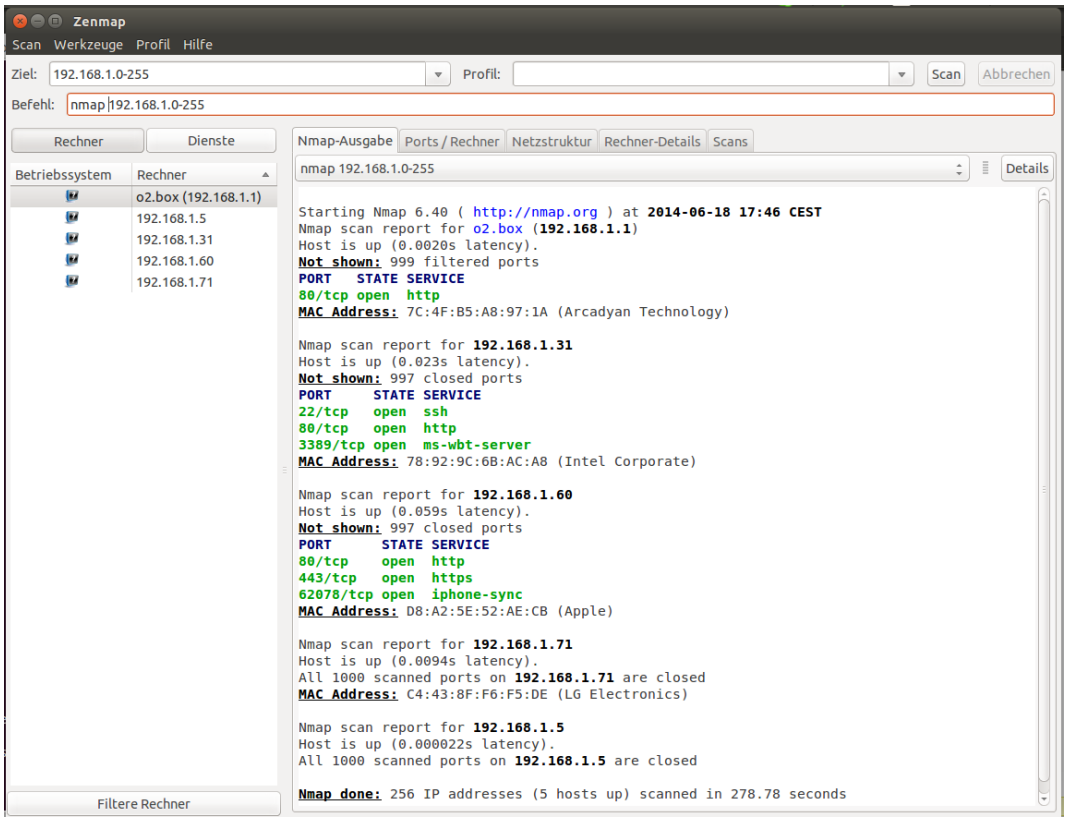

Abbildung 2.2: Oberfläche von Zenmap

Alle Informationen, die Zenmap zusätzlich ausgibt, können auch über die Konsole mit den entsprechenden Einstellungen eingeholt werden.

[3] Zenmap [zenmap]

2.4 Film und Fernsehen

Nmap wird in diversen Filmen genutzt. So hackt sich in “Matrix Reloaded” der Charakter Trinity in ein Kraftwerk, nachdem der Rechner mit Nmap gescannt wurde. In „Bourne Ultimatum“ wird die damalige Beta-Version 4.01 mit Zenmap benutzt. Im Film „Elysium“ wird eine Zukunftsversion von Nmap genutzt. Hier wird Nmap mit der Version 13 genutzt, obwohl aktuell (Stand 10/2014) erst die Version 6.4 existiert.

3 ZMap

ZMap wurde 2013 an der University of Michigan entwickelt, um internetweite Langzeitstudien durchzuführen, die mit Nmap aufgrund der limitierten Geschwindigkeit nicht möglich sind.

3.1 Softwarearchitektur

Für ZMap wurde ein sehr modulares Design umgesetzt, um viele Arten der Untersuchung zu unterstützen. ZMap kann man grob in drei Teile unterteilen. Zum Scanner Core gehören die Module „Configuration, Adressing and Timing“, „raw socket“ und „libpcap“. Das Probe Module besteht aus den Teilen „Probe Generation“ und „Response Interpretation“. Der „Output Handler“ ist ein eigenes Modul.

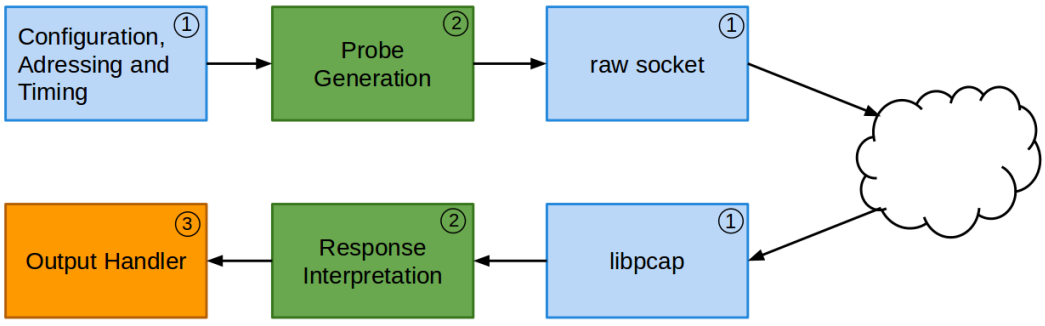

Abbildung 3.1: Schema-Darstellung der Softwarearchitektur von ZMap[1]

1. Scanner Core (blau)
   Der Scanner Core setzt sich aus verschiedenen Bestandteilen zusammen. Er ist zuständig für das Parsen der Kommandozeile und das Auslesen von Konfigurationsdateien. Außerdem wird hier die Generierung sowie der Ausschluss von Adressen realisiert. Zudem ist der Scanner Core für das Lesen und Schreiben von Netzwerk-Paketen zuständig.
2. Probe Module (grün)
   Das Probe Module füllt den Body eines Paketes und validiert die einkommenden Pakete und verwirft sie gegebenenfalls. Die Modularität von ZMap erlaubt die Unterstützung von vielen verschiedenen Untersuchungsmethoden und Protokollen. Generell unterstützt ZMap TCP Port Scanning sowie ICMP Echo Scanning. Durch die Implementierung weniger Callback-Funktionen ist es jedoch auch möglich eigene Scan-Typen hinzuzufügen.
3. Output Handler (orange)
   Der Output Handler ist für die Ausgabe der Ergebnisse zuständig. Dies kann auf unterschiedliche Art und Weise geschehen. Die Ergebnisse können einfach auf die Kommandozeile ausgegeben oder in eine Datenbank geschrieben werden. Es ist auch möglich die Ergebnisse an andere Prozesse weiterzugeben.

3.2 Funktionsweise

ZMap arbeitet stateless nach dem Prinzip „Send and Forget“. Im Gegensatz zu Nmap muss durch dieses Verfahren keine Liste von ausstehenden Antworten gehalten werden. Standardmäßig macht ZMap einen TCP SYN Scan auf einen spezifizierten Port mit der maximal möglichen Rate an Pakete pro Sekunde.

Adressierung der Pakete

Die Adressierung der Pakete geschieht nicht in der numerischen Reihenfolge der Adressen, da es so möglich wäre, dass ein Netzwerk zu viel Traffic erhält und nicht alle Anfragen beantwortet. Hierdurch würde das Ergebnis eines Scans verfälscht. Stattdessen wird nach einer zufälligen Permutation des Adressraums gescannt. Es wird eine multiplikative Gruppe von Integern modulo p gebildet, wobei p eine Primzahl ist, die ein wenig größer als 2^{32} ist. p muss eine Primzahl sein, damit die Gruppe zyklisch ist und alle Adressen erreicht werden können. Für jeden Scan soll eine neue Permutation entstehen. Dies wird erreicht, indem jedes Mal eine neue Primzahl und eine zufällige Startadresse gewählt wird. Um einen eingeschränkten Satz zufälliger Adressen zu scannen, wird einfach eine Teilmenge der kompletten Permutation genutzt. Bei der Adressierung muss auch darauf geachtet werden, dass die über die Blacklist oder Kommandozeile spezifizierten Adressen ausgeschlossen werden, damit

[1] Vgl. [ZMapSlides] (Folie 14)

sie nicht mitgescannt werden. Dies wird über einen Radix-Tree realisiert, der speziell für Intervalle designed wurde und häufig für Routing-Tabellen genutzt wird.

Paket-Übertragung und -Empfang (Packet Transmission and Receipt)

ZMap sendet Anfrage so schnell es CPU und NIC (Network Interface Controller) erlauben. Die Komponente, die die Pakete generiert, arbeitet asynchron mit mehreren Threads. Jeder Thread hat eine (geschlossene) Schleife, die Ethernet-Layer-Pakete über ein Raw Socket sendet. Der Ethernet Layer wird verwendet, um bestimmte Werte aus den Paketen zu cachen und somit Kernel Overhead zu vermeiden. So ändert sich im Ethernet-Header während eines Scans, abgesehen von der Checksumme, nichts und muss deshalb auch nicht jedes Mal neu erstellt werden. Die Pakete werden über ein Raw Socket versendet und empfangen. Dies hat den Vorteil, dass das Senden und Empfangen von Paketen nicht über einen bestimmten Transport Layer und dessen Protokoll-Vorgaben geschehen muss. Da keine komplette TCP-Verbindung aufgebaut wird, kann auf ein eingehendes TCP SYN-ACK-Paket automatisch mit TCP RST geantwortet werden und somit die Connection wieder geschlossen werden. Als Empfangskomponente wurde die Bibliothek libpcap genutzt. pcap (packet capture) ist eine freie API, um Netzwerktraffic zu erfassen. Mithilfe von libpcap können Pakete gefiltert und beispielsweise in eine Datei gespeichert werden, um sie mit anderen Programmen, wie in diesem Fall ZMap, auszuwerten. Da libpcap jedes einkommende Paket untersuchen muss, kann dies die Performance von ZMap einschränken. Jedoch wird nur ein Bruchteil der ausgehenden Pakete beantwortet, da meist ein Großteil der Hosts nicht erreichbar sind. Von den eingehenden Paketen wird die Quell- und Zieladresse überprüft und dabei die Pakete verworfen, die eindeutig nicht zu diesem Scan gehören. Die restlichen Pakete werden zur Interpretation an das Probe Module geschickt.

Erstellen eines Pakets

Der Scanner Core stellt einen leeren Buffer für die Pakete zur Verfügung und das Probe Module füllt ihn mit statischen Daten, die für alle Ziele gleich sind. Für jeden Host, der gescannt werden soll, werden außerdem noch Host- und Scan-spezifische Daten in veränderbare Datenfelder des Paketes geschrieben. Die Felder, die hierfür genutzt werden, müssen einen Effekt auf Felder im Antwort-Paket haben. Dies funktioniert so ähnlich wie SYN Cookies, die bei einer Anfrage an den Client gesendet werden und bei einer Antwort vom Server ausgelesen werden können. Für jeden zu scannenden Host wird ein MAC (Message Authentication Code) berechnet und in ein beliebiges Feld geschrieben. Aus Performance-Gründen wird bei ZMap der UMAC (Message Authentication Code, der auf Universal Hashing basiert) genutzt. Bei einem TCP Port Scan wird der Quell-Port und eine Squenznummer genutzt. Bei einem ICMP Scan wird der ICMP Indentifier und eine Sequenznummer genutzt.

Testen, ob Paket eine Antwort ist im Probe Module

Die libpcap-Bibliothek kann bei einkommenden Paketen nur grob prüfen, ob sie zum Scan gehören. Im Probe Module werden die beschriebenen Felder geprüft und so Hintergrundtraffic und Antworten auf frühere Scans aussortiert.

Ausgabe der Antworten

Im Output Module wird entschieden, ob die Scan-Resultate einfach ausgegeben oder weiterverarbeitet werden sollen. In der Implementierung von ZMap besteht die Ausgabe aus einem einfachen Text mit einer Liste von IP-Adressen oder aus einem erweiterten Text, der eine Liste von allen Antwort-Paketen und Zeitangaben beinhaltet. Es ist auch möglich, ein Interface zu nutzen um die Resultate zum Beispiel in einer Redis In-Memory-Datenbank zu sammeln. Das Output Module kann dahingehend erweitert werden, dass Netzwerk-Events getriggert werden, beispielsweise um ein Handshake abzuschließen. Hierbei ist es am einfachsten eine neue TCP-Connection mit der antwortenden Adresse aufzubauen. Dies kann auch asynchron passieren. Um sofort nach Erhalt der positiven Antwort die TCP-Verbindung aufzubauen, kann das Kernel Modul forge_socket genutzt werden. Durch diese Erweiterung ist es möglich die Session-Parameter zu speichern und den Handshake mit der initialen Anfrage zu komplettieren.

3.3 Probleme

Ein Nachteil von ZMap ist, dass es nur für Linux implementiert wurde. Zudem gibt es aktuell noch keine GUI. Bei einer schlechten Reihenfolge der zu scannenden Adressen kann es außerdem zu einer Überlastung fremder Netzwerke kommen. Ein weiteres Problem ist, dass IPv6-Adressen nicht gescannt werden können, da der Adressraum von IPv6 zu groß ist. ZMap kann nicht einfach erweitert werden, sondern es ist notwendig eine neue Methode zum Scannen von IPv6-Adressen zu entwickeln.

3.4 Vergleich mit Nmap

Ein Vergleich mit Nmap ist nicht leicht, da die beiden Tools, zwar beide Internetscanner sind, aber zu unterschiedlichen Zwecken entwickelt wurden.

	Nmap	ZMap
Zweck	vielseitig, testen von vielen Ports von geringer Zahl von Hosts	testen eines Ports von sehr vielen Hosts
Architektur	hält Liste mit ausstehenden Responses	stateless; send and forget
Geschwindigkeit um das Internet zu scannen[2]	62,5 Tage	1:09 Stunden
GUI	Zenmap	-
Betriebssystem	Linux, Windows	Linux

Tabelle 3.1: Vergleich von Nmap und ZMap

[2] Hierbei wurde 1 Million zufällige IPv4-Adressen gescannt und das Ergebnis auf alle IPv4-Adressen des Internets hochgerechnet [ZMapSlides] (Folie 18)

4 Projekte

Auf der offiziellen Homepage von ZMap werden verschiedene Projekte[1] vorgestellt, die den Internetscanner nutzen. Einige dieser Projekte werden im Folgenden vorgestellt.

4.1 Heartbleed Bug

Der Heartbleed-Bug wurde im April 2014 entdeckt. Es ist ein Fehler in der Open-Source-Bibliothek OpenSSL, durch den private Daten, unter anderem auch Passwörter in Klarschrift, ausgelesen werden konnten. Das Ziel dieser Untersuchung war, zu tracken, welche Hosts betroffen sind und wie schnell Updates gemacht wurden[2]. Hierfür wurden "Alexas Top 1 Million Domains"[3] gescannt.

- 24 - 55% der populären HTTPS-Seiten waren betroffen
- zwei Tage später waren noch immer 11% anfällig
- nach zwei Monaten hatten 3% die Sicherheitslücke noch nicht durch ein Update geschlossen

[1] Projects using ZMap [Projekte]
[2] Heartbleed Report [MatterOfHeartbleed] and [HeartbleedHealthReport]
[3] `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip`

4.2 Hurricane Sandy

Der Hurrikan Sandy zog in der Zeit vom 29.10. bis 31.10.2014 über die Ostküste der USA. Mithilfe von ZMap sollten die Auswirkungen des Sturms sichtbar gemacht werden. Hierzu wurden zunächst IP-Adressen geografisch lokalisiert, um in den folgenden Scans nur die Adressen anzusprechen, die in der vorhergesagten Bahn des Hurrikan waren. Danach wurden drei Tage lang alle zwei Stunden die Adressen auf dem Port 443 gescannt, um zu sehen wann welcher Host nicht mehr antwortet. In die Übersicht, welche Orte besonders vom Sturm beeinträchtigt waren, wurde die Orte aufgenommen, in denen mehr als 30% weniger Hosts antworteten. Auf der Karte sind als am meisten betroffene Gebiete New Jersey, die New York City sowie die Halbinsel Long Island zu sehen.

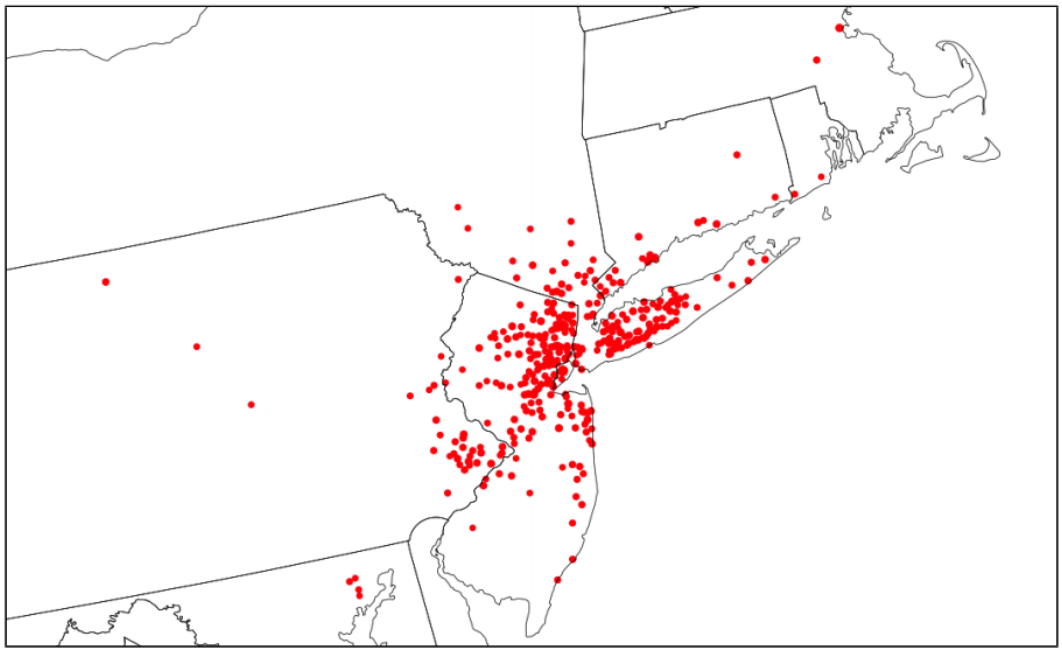

Abbildung 4.1: Orte, in denen mehr als 30% weniger antwortenden Hosts[4]

[4] Vgl. [ZMapPaper] Seite 11

4.3 The Internet's Sleep Cycle

Um herauszufinden, wann die beste Zeit ist um das Internet zu scannen, wurden Scans zu unterschiedlichen Tageszeiten gemacht. Die Annahme der Autoren war, dass bei viel Traffic Pakete leichter verloren gehen, also weniger Hosts antworten und somit der Scan weniger aussagekräftig ist. Ausgehen vom Standort Michigan scheinen die besten Zeiten um Scans zu machen der frühe Morgen zu sein, während der schlechteste Zeitpunkt der frühe Abend ist.

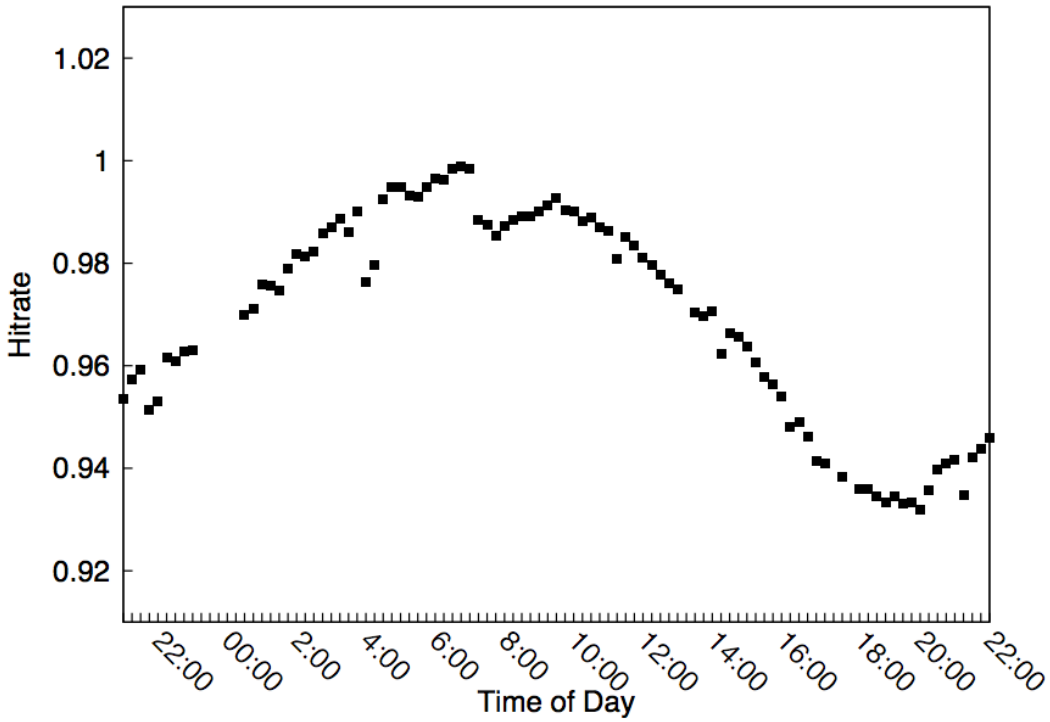

Abbildung 4.2: The Internet's Sleep Cycle[5]

[6]

4.4 Mining Your P's and Q's: Detection of Widespread Weak Keys in Network Devices

Dieses Projekt hatte das Ziel die Sicherheit der RSA- und DSA-Verschlüsselung im Internet zu prüfen. Hierbei wurde ein Scan auf TLS- und SSH-Server gemacht und

[6] [ZMapPaper] Seite 5

die Keys überprüft. Eine RSA- oder DSA-Verschlüsselung wird unsicher, wenn der Zufallszahlengenerator nur Pseudo-Zufallszahlen generiert. Dies bedeutet, dass die Zufallszahlen in einer nachstellbaren Reihenfolge erzeugt werden. Bei einem Scan des IPv4 Adress-Bereichs wurden 5,8 Millionen TLS-Zertifikate von 12,8 Millionen Hosts und 6,2 Millionen SSH-Host-Keys von 10,2 Millionen Hosts gesammelt. Die Ergebnisse waren nicht sehr erfreulich[7]:

- 5,57% der TLS-Hosts und 9,6% der SSH-Hosts benutzten die gleichen Schlüssel wie andere Hosts
- mindestens 5,23% der TLS-Hosts benutzen Default-Schlüssel, die nie geändert wurden
- weitere 0,34% haben die gleichen Schlüssel wie mindestens ein anderer Hosts generiert
- bei 0,5% (64.000) der TLS-Hosts und 1,06% (108.000) der SSH-Hosts war es möglich die privaten Schlüssel zu ermitteln.

4.5 Hunting Botnets with ZMap

Bei dieser Untersuchung sollte herausgefunden werden, wie viele Hosts mit Botnets infiziert sind. Hierfür musste ein Payload geschrieben werden, durch den man herausfinden kann, ob ein Host infiziert ist. Es wurden fünf verschiedene Ports gescannt mit folgendem Ergebnis[8]:

Port	Anzahl der infizierten Hosts
16461	239
16464	3.503
16465	1.285
16470	2.192
16471	4.230
Insgesamt infizierte Hosts	10.500

Tabelle 4.1: Liste der Ports und infizierten Hosts

[7] Mining Your P's and Q's [MiningYourPsandQs]
[8] Hunting Botnets [HuntingBotnets]

5 Fazit

ZMap ist ein Tool, mit dem internetweite (Langzeit-) Studien leicht zu machen sind. Im Gegensatz zu Nmap, kann der IPv4-Adressraum des Internets leicht und schnell gescannt werden. Hierzu muss allerdings auch gesagt werden, dass im Gegensatz zu Nmap, nicht für jeden Host viele Informationen gesammelt werden, sondern jeder Scan hat einen bestimmten eingegrenzten Zweck, für den das Programm oder eventuelle Erweiterungen speziell implementiert wurden.

5.1 Scanning Best Practices

Wichtig bei der Nutzung von ZMap ist es, gewisse Regeln zu beachten, auf die die Entwickler auch selbst hinweisen[1].

> „We offer these suggestions for researchers conducting Internet-wide scans as guidelines for good Internet citizenship.
>
> 1. Coordinate closely with local network administrators to reduce risks and handle inquiries
> 2. Verify that scans will not overwhelm the local network or upstream provider
> 3. Signal the benign nature of the scans in web pages and DNS entries of the source addresses

[1] Scanning Best Practices [BestPractices]

4. Clearly explain the purpose and scope of the scans in all communications
5. Provide a simple means of opting out and honor requests promptly
6. Conduct scans no larger or more frequent than is necessary for research objectives
7. Spread scan traffic over time or source addresses when feasible

It should go without saying that scan researchers should refrain from exploiting vulnerabilities or accessing protected resources, and should comply with any special legal requirements in their jurisdictions.“

Literaturverzeichnis

[ZMap] University of Michigan. *ZMap - The Internet Scanner.* https://zmap.io/index.html [12.11.2014]

[ZMapPaper] Durumeric, Zakir; Wustrow, Eric; Halderman, J. Alex (August 2013). *ZMap: Fast Internet-Wide Scanning and its Security Applications.* https://zmap.io/paper.pdf [11.11.2014]

[ZMapSlides] Durumeric, Zakir; Wustrow, Eric; Halderman, J. Alex (August 2013). *ZMap: Fast Internet-Wide Scanning and its Security Applications.* https://zmap.io/zmap-talk-sec13.pdf [12.11.2014]

[Projekte] University of Michigan. *Projects using ZMap.* https://zmap.io/projects.html [11.11.2014]

[HeartbleedHealthReport] University of Michigan. *Heartbleed Bug Health Report.* https://zmap.io/heartbleed/ [11.11.2014]

[MatterOfHeartbleed] Durumeric, Zakir; Kasten, James; Adrian, David; Halderman, J. Alex; Bailey, Michael (Oktober 2014). *The Matter of Heartbleed.* https://jhalderm.com/pub/papers/heartbleed-imc14.pdf [11.11.2014]

[MiningYourPsandQs] Heninger, Nadia; Durumeric, Zakir; Wustrow, Eric; Halderman, J. Alex (August 2012). *Mining Your P's and Q's: Detection of Widespread Weak Keys in Network Devices.* https://factorable.net/paper.html [11.11.2014]

[HuntingBotnets] Lawshae, Ricky (2014). *Hunting Botnets with ZMap.* http://h30499.www3.hp.com/t5/HP-Security-Research-Blog/Hunting-Botnets-with-ZMap/ba-p/6320865#.VGIijIVEZVa [11.11.2014]

[BestPractices] University of Michigan. *Scanning Best Practices.* https://zmap.io/documentation.html#bestpractices [12.11.2014]

[DataRespository] University of Michigan. *Internet-Wide Scan Data Repository.* https://scans.io/ [18.11.2014]

[ACM] ACM Digital Library. *ACM Digital Library.* http://dl.acm.org/citation.cfm?id=2534818 [18.11.2014]

[SecurityBlog] Dalziel, Henry. *Using ZMap, scan the ENTIRE Internet in 44 Minutes.* http://www.concise-courses.com/security/zmap-scanning/ [18.11.2014]

[Portscans] *Portscans.* http://www.tcp-ip-info.de/security/portscans.htm [18.11.2014]

[IranianHacker] University of Michigan. *Report of Iranian hackers taking over U-M computers called pure fiction.* http://www.ur.umich.edu/update/archives/121017/cyber [18.11.2014]

[IranStrikesBack] Gertz, Bill; The Washington Free Beacon (15.10.2012). *Iran Strikes Back.* http://freebeacon.com/national-security/iran-strikes-back/ [18.11.2014]

[ipv6hackers] Gont, Fernando (04.9.2013). *Zmap.* http://lists.si6networks.com/pipermail/ipv6hackers/2013-September/001352.html [18.11.2014]

[nmap] Lyon, Gordon. *Nmap Security Scanner.* http://nmap.org/ [18.11.2014]

[nwlab] Kulpa, Mirko. *Nmap Tutorial.* http://www.nwlab.net/tutorials/portscanner/nmap-tutorial.html [18.11.2014]

[selflinux] SelfLinux. *nmap - der Netzwerksicherheits-Scanner.* http://www.selflinux.org/selflinux/html/nmap.html [18.11.2014]

[zenmap] Netzwerk-Tools. *Zenmap.* http://www.netzwerk-tools.com/Zenmap.php [18.11.2014]

[LinuxJournal] LJ Staff; Linux Journal (01.12.2001). *Editors' Choice Awards.* http://www.linuxjournal.com/article/5525 [18.11.2014]